

修士論文の和文要旨

研究科・専攻	大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程		
氏 名	田中 亮輔	学籍番号	1631090
論文題目	通信路変動に対するロバスト性を有した符号化協調通信に関する研究		

要 旨

近年、身の回りのあらゆるものがネットワークに繋がる、モノのインターネット (IoT: Internet of Things) が注目を集めている。IoT では環境中に無数に配置された端末が環境情報を取得し、基地局に送信することによって実現される。しかし無線通信においては、フェーディングやシャドウイングと呼ばれる物理現象によって受信信号電力そのものが確率的に変動し、安定した通信を保証することができない。そこでダイバーシティ利得に加えて、符号化利得も同時に得ることができる符号化協調通信が提案されている。これは各端末が共有し合った情報からパリティを生成し、基地局に送信することで、端末間で分散的に誤り訂正符号を構成する方式である。

近年、符号化協調通信の一方式として低演算量で符号化が可能で、特に優れた復号特性をもつ空間結合 Repeat-Accumulate(SC-RA: Spatially Coupled Repeat-Accumulate) 符号に基づいた、SC-RA 符号化協調 (SC-RA-CC: SC-RA Coded Cooperation) が提案されている。これは、複数の RA 符号をプロトグラフと呼ばれるグラフ上で結合することにより構成されている。

一方、空間結合符号では通信路上でバースト誤りが発生した場合、確率伝搬法によるプロトグラフ上での復号の連鎖が停止し、誤り率特性が劣化することが報告されている。この問題に対して、プロトグラフを多次元で構成することにより、復号の連鎖が停止する確率を小さくする多次元空間結合符号が提案されている。この符号構造を SC-RA-CC に適用することにより、バースト誤りに対するロバスト性を付与した多次元 SC-RA-CC(MD-SC-RA-CC: Multi-Dimensional SC-RA-CC) が提案されている。しかし MD-SC-RA-CC については、消失通信路における解析のみであり、バースト誤りが発生する環境での誤り率特性は評価されていなかった。また MD-SC-RA-CC の符号化法についても十分な議論がなされていなかった。

そこで、本研究ではバースト誤りが発生する通信路において、優れた復号特性を示す MD-SC-RA-CC の結合方法を提案する。まず、高 SNR(Signal-to-Noise Ratio) 領域におけるブロックレイリーフェーディング通信路を模倣したブロック消失通信路 (BEC: Block Erasure Channel) を仮定する。BEC 上での MD-SC-RA-CC の復号失敗確率の下界を導出し、下界に漸近する符号化法が持つべき設計指針を求める。BEC 上においてその設計指針を満たす符号化法を探索し、それが優れた復号特性を有することを示す。最後に計算機シミュレーションを用いて、ブロックレイリーフェーディング通信路における上記の符号化法の誤り率特性を評価し、提案方式が優れていることを示す。

平成29年度 修士論文

通信路変動に対するロバスト性を有した

符号化協調通信に関する研究

学籍番号 1631090

氏 名 田中 亮輔

指導教員 石橋 功至 准教授

電気通信大学 情報理工学研究科

情報・ネットワーク工学専攻

提出日 平成30年3月13日

概要

無線通信ではフェーディングやシャドウイングの影響により、受信信号電力が確率的に変動するため、いかなる誤り訂正符号を用いても誤り確率を0にすることができない。そこで符号化利得に加えて、ダイバーシティ利得を得ることができる手法として、符号化協調通信が提案されている。これはネットワーク上の全ての端末がお互いに情報を共有し合い、1つの誤り訂正符号を分散的に構成することによって実現される。符号化協調通信の一方式として、符号化演算量が小さく、シャノン限界に漸近する復号特性を有する空間結合 Repeat-Accumulate (SC-RA: Spatially coupled) 符号をネットワーク上で構成した、SC-RA 符号化協調 (SC-RA-CC: SC-RA Coded Cooperation) が提案されている。しかし低速フェーディング通信路のようにバースト誤りが発生する環境では、空間結合符号の構造により復号特性が劣化することが報告されている。そこで、SC-RA-CC の符号構造を多次元化することにより、バースト誤りに対するロバスト性を付加した多次元 SC-RA-CC (MD-SC-RA-CC: Multi-Dimensional SC-RA-CC) が提案されている。しかし先行研究においては、消失通信路での解析のみであり、バースト誤りが発生する環境での特性は評価されていなかった。また MD-SC-RA-CC の符号構成法についても十分な議論がなされていなかった。本稿では、ブロックフェーディング通信路において、優れた復号特性を示す MD-SC-RA-CC の符号化法を提案する。そして、その誤り率特性を計算機シミュレーションを用いて評価することにより、提案方式が優れていることを示す。

目 次

第 1 章	序論	2
1.1	研究背景	2
1.2	本論文の構成	4
第 2 章	システムモデル	5
第 3 章	多次元空間結合 Repeat-Accumulate 符号化協調	8
3.1	Repeat-Accumulate 符号	8
3.1.1	符号化法	8
3.1.2	グラフ表現	10
3.1.3	復号法	12
3.2	空間結合 RA 符号	19
3.2.1	符号化法	19
3.2.2	プロトグラフ表現	22
3.2.3	復号法	26
3.3	多次元空間結合 RA 符号化協調	26
3.3.1	符号化法	26
3.3.2	パリティ検査行列	28
3.3.3	グラフ表現	29
3.3.4	符号化率	31
第 4 章	符号化協調通信の符号化設計	35
4.1	ブロック消失通信路モデル	35
4.2	密度発展法	35
4.3	復号失敗確率の下界	37
4.3.1	1-BEC の場合	37
4.3.2	T -BEC の場合	37
4.4	符号化構造の設計指針	38
4.4.1	1-BEC の場合	38
4.4.2	T -BEC の場合	39
4.5	BEC における数値例	39
4.5.1	1-BEC の場合	40
4.5.2	T -BEC の場合	41
4.6	パンクチャド MD-SC-RA-CC	43
4.6.1	パンクチュアリングパターン	43
4.6.2	1-BEC の場合	44
4.6.3	T -BEC の場合	44
第 5 章	数値結果	47

第 6 章 結論	50
謝辞	51
関連発表	52

図 目 次

2.1	本研究において想定するネットワーク	6
2.2	ネットワーク内の各端末が従う送信スケジュール	7
3.1	RA 符号化器 (\mathcal{D} は 1bit 遅延器 (Delay Device), \oplus は排他的論理和演算子)	9
3.2	(Q, a) -regular RA 符号のタナーグラフ表現	11
3.3	3 変数ファクターグラフの一例	12
3.4	メッセージ交換による周辺化計算の一例	14
3.5	(Q, a, L) -空間結合 RA 符号の符号化器の構造	20
3.6	(Q, a) -regular RA 符号のプロトグラフ表現	23
3.7	(Q, a) -regular RA 符号のプロトグラフを K 個コピーし並べたもの	24
3.8	空間結合 RA 符号のプロトグラフ表現	25
3.9	スロット j における端末 i の符号化器 (ただし, $\mathcal{C}(j) = \{c_1, c_2, \dots, c_{ \mathcal{C}(j) }\}$)	27
3.10	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC のパリティ検査行列	28
3.11	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 9$ である MD-SC-RA-CC のパリティ検査行列	29
3.12	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC の 1 次元プロトグラフ表現	30
3.13	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 9$ である MD-SC-RA-CC の 1 次元プロトグラフ表現	31
3.14	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC の 2 次元プロトグラフ表現	33
3.15	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 9$ である MD-SC-RA-CC の 2 次元プロトグラフ表現	34
4.1	全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC の 1 次元プロトグラフ表現. スロット 4, 5 の情報ノードと, スロット 5, 6 のチェックノード間において長さ 4 のループ, スロット 12, 13, 14 の情報ノードとスロット 14, 15, 16 のチェックノード間において長さ 6 のループが存在	38
4.2	1-BEC における結合ベクトル $\mathbf{v} = (2, 6, 8, 9), (1, 2, 3, 4), (1, 3, 4, 9)$ の MD-SC-RA-CC の DFP 特性 ($N = 5, T = 20, Q = 4$)	40
4.3	1-BEC における結合ベクトル $\mathbf{v} = (2, 6, 8, 9), (1, 2, 3, 4), (1, 3, 4, 9)$ のパンクチャード MD-SC-RA-CC の DFP 特性	44

5.1	1-ブロックレイリーフェーディング通信路における結合ベクトル $\mathbf{v} = (1, 3, 4, 9), (2, 6, 8, 9), (1, 2, 3, 4)$ であるパンクチャド MD-SC-RA-CC と SC-RA-CC の PER 特性	48
5.2	T -ブロックレイリーフェーディング通信路における結合ベクトル $\mathbf{v} = (1, 3, 4, 9), (2, 6, 8, 9), (1, 2, 3, 4)$ であるパンクチャド MD-SC-RA-CC と SC-RA-CC の PER 特性	49

第1章 序論

1.1 研究背景

近年、身の回りのあらゆるものがネットワークに繋がる、モノのインターネット (IoT: Internet of Things)[1] が注目を集めている。IoT では環境中に無数に配置された端末が環境情報を取得し、基地局に送信することによって実現される。しかしながら、IoT において使用される端末は無数に配置されるため、小型で安価であるという制約を受けることになる。そのため、端末のバッテリーサイズは小型になり、端末の送信信号電力はできる限り小さく抑える必要がある。また経済的観点から、複数のアンテナを具備することが難しい。端末はこのような制約を持つ一方、無線通信においては、フェーディングやシャドウイングと呼ばれる物理現象によって受信信号電力そのものが確率的に変動するため、安定した通信を保証することができない。この問題を端末側で解決する手段として、送信ダイバーシティ技術の1つである時空間ブロック符号 (STBC: Space Time Block Code)[2] が Alamouti により提案されている。これは符号化された信号を端末に具備された複数のアンテナから送信し、受信側ではその信号を時間軸上で分離する方式である。これによりダイバーシティ効果を得ることができ、安定した通信を行うことが可能になる。しかしながら、この方式はIoT においては経済的観点から現実的ではない。

これらの要求に対して、協調通信技術 [3] が提案されている。これは情報を送信する端末の近傍に他の端末が存在するとき、その端末を協力端末とみなして情報パケットを共有し、同一の情報パケットを複数の端末から送信する方式である。協調通信には、協力端末が受信した信号をそのまま増幅して送信する増幅・転送 (Amplify-and-Forward) 協調 [4] や、復調・復号を行い送信する復号・転送 (Decode-and-Forward) 協調 [5, 6] が存在する。これらの方式ではダイバーシティ効果により、通信路の変動による影響を緩和し安定した通信を実現することができる。一方、同一の情報パケットを複数の端末で共有し送っているため、これは単純な繰り返し符号と考えることができ、帯域利用効率という観点では非常に無駄が多く、さらに符号そのものによる誤り訂正能力はほとんど期待することができない。

そこで各端末が共有した情報パケットからパリティパケットを生成し、それを基地局に送信することにより、端末間で分散的に誤り訂正符号を構成する符号化協調通信 [7, 8] が提案されている。この方式では、協調通信によるダイバーシティ利得に加えて、誤り訂正符号による符号化利得も同時に得ることができるという利点がある。符号化協調で分散的に生成する誤り訂正符号については、様々なものが提案されてきた。シャノンの通信路容量に迫る特性を示すターボ符号を用いた符号化協調通信 [9] や、高速フェーディング通信路において優れた誤り率特性を示す時空間ブロック符号化協調 [9] などがある。しかし前者の方式では、端末数が2台のみという制約を受け、後者の方式では端末数が増大したときに符号を構成するためのオーバーヘッドが大きくなるという問題があった。

そこで端末数が大きい状況においても、オーバーヘッドが小さく符号化協調通信を実現することができる方式として、適応ネットワーク符号化協調 (ANCC: Adaptive Network Coded Cooperation)[10] が提案されている。この方式の通信プロトコルは、ブロードキャストとリレーの 2 フェーズに分かれており、まずブロードキャストフェーズにおいて全ての端末が自身の情報パケットをブロードキャストしネットワーク全体で共有を行う。次に、リレーフェーズにおいて共有された情報パケットを決められた数だけ選択し、それらの排他的論理和をとることでパリティパケットを生成し、それを基地局に送信する。ANCC を実現するための前処理としては、リレーフェーズにおいて各端末が選択するパケットの個数を共有することのみである。ANCC において構成される誤り訂正符号は、1960 年代に Gallager によって提案された低密度パリティ検査 (LDPC: Low-Density Parity Check) 符号 [11] の一種の、低密度生成行列 (LDGM: Low-Density Generate Matrix) 符号であり、非常に疎な検査行列により定義される。しかし、ANCC の検査行列内には長さ 4 のショートループが高い確率で発生するため、復号特性は高い誤り率においてエラーフロアを示す [12]。この問題に対して、ANCC を改良した方式が文献 [13] において提案されている。これは符号化協調通信に先駆けて、長さ 4 のショートループがない検査行列を生成し、それを全ての端末と共有する方式である。この方式により優れた誤り率特性を示す一方、符号化協調通信を行うためのオーバーヘッドは大きくなり、ANCC の利点を損なう。

近年、低演算量で符号化が可能で、特に優れた復号特性をもつ空間結合 Repeat-Accumulate(SC-RA: Spatially Coupled Repeat-Accumulate) 符号 [14] が提案されている。SC-RA 符号の空間結合符号は、1999 年に文献 [15] において発表された畳み込み LDPC 符号から発展したものである。畳み込み LDPC 符号は統計的に独立な複数の LDPC 符号に対して、遅延器を用いた畳み込み処理を行うことによって構成される組織符号である。この畳み込み符号に対して終端処理を行った符号が空間結合 LDPC 符号と呼ばれる。この符号の特徴としては、畳み込み処理の開始時と終了時には、符号化率の低いパリティパケットが生成され、それ以外の処理時には符号化率の高いパリティパケットが生成されることにある。よって、確率伝搬 (BP: Belief Propagation) 法 [16] に基づく復号では、符号化率の低い開始時と終了時に生成されたパリティパケットから開始される。繰り返し復号を行うごとに、この誤り訂正が隣接する符号に広がり、さらに連鎖的に符号全体に拡散していく。この符号構造により、空間結合符号の最大の利点である、閾値飽和 [17] という現象が発生する。これは空間結合符号の BP 閾値が、それを構成する単一の符号の MAP(Maximum A Posteriori) 閾値に漸近的に一致するというものである。もう 1 つの特徴としては、あらゆる通信路環境において、シャノンの通信路容量に漸近するユニバーサル性を持つ符号であることが証明されていることである [18]。

次に、SC-RA 符号を構成する Repeat-Accumulate(RA) 符号 [19] は LDPC 符号の一種であり、疎な検査行列に基づく優れた復号特性をもつ。RA 符号の符号化器は、単純なビット繰り返しと、作動符号化器のみによって構成されるため、LDPC 符号と比較して低演算量かつ低遅延で符号化が可能である。以上から、SC-RA 符号は LDPC 符号に変わって、RA 符号を空間結合させた符号であると言える。

そして、この符号に基づいた符号化協調通信として、SC-RA 符号化協調 (SC-RA-CC: SC-RA Coded Cooperation)[20] が提案されている。SC-RA-CC の利点は、低演算な符号化と優れた復号特性に加え、各端末が送信する情報パケットが短い場合においても、複数の情報パケットを畳み

込むために、端末全体では効果的に長い符号語を構成できることである。これにより優れた復号特性を得ることができる。

しかしながら、通信路上で誤り確率の高い状況が一定期間連続するバースト誤りが発生した場合、BP 法による空間結合符号の復号の連鎖が停止し、誤り率特性が劣化することが報告されている [21]。無線通信の観点で考えたとき、バースト誤りのような環境は、フェージングによって受信信号電力が長期間落ち込んだときに対応する。そこで文献 [21] では、畳み込み方法を工夫することにより、符号化率が低いパリティパケットを数多く生成し、BP 法による復号の連鎖が停止しにくい構造をもつ、多次元空間結合符号が提案されている。そして、この符号構造を SC-RA-CC に適用することにより、バースト誤りに対するロバスト性を付与した多次元 SC-RA-CC(MD-SC-RA-CC: Multi-Dimensional SC-RA-CC) が提案されている [22]。文献 [22] では、消失通信路における解析のみであり、バースト誤りが発生する環境での誤り率特性は評価されていなかった。また MD-SC-RA-CC の符号構成法についても十分な議論がなされていなかった。

そこで、本研究ではバースト誤りが発生する通信路において、優れた復号特性を示す結合方法を提案する。まず、高 SNR(Signal-to-Noise Ratio) 領域におけるブロックレイリーフェーディング通信路を模倣したブロック消失通信路 (BEC: Block Erasure Channel) を仮定する。BEC 上での MD-SC-RA-CC の復号失敗確率の下界を導出し、下界に最も漸近する符号化法を探索する。最後に計算機シミュレーションを用いて、ブロックレイリーフェーディング通信路における上記の符号化法の誤り率特性を評価し、提案方式が従来方式よりも優れていることを示す。

1.2 本論文の構成

本論文は次のように構成される。第 2 章では、本研究で扱うシステムモデルについて述べ、通信路モデルを定義する。第 3 章では、多次元空間結合 RA 符号化協調通信の符号化法と、行列表現、グラフ表現について説明する。第 4 章では、簡易な通信路を導入し、多次元空間結合 RA 符号化協調の符号化設計について議論する。第 5 章では、最適化された多次元空間結合 RA 符号化協調の符号化方法について、計算機シミュレーションにより評価を行い、第 6 章で本論文を総括する。

第2章 システムモデル

本研究において想定するネットワーク構成を図 2.1 に示す.

ネットワーク上には, N 台の情報源端末と 1 つの共通の基地局が存在すると仮定する. 各端末は一本のアンテナを具備しており, 半二重通信の制約のもと時分割多元接続 (TDMA: Time-Division Multiple Access) によって通信を行う. 各端末は, サイズ M [bits] の統計的に独立な T 個の情報パケットを持っているとする.

図 2.2 に, ネットワーク内の端末が従う送信スケジュールを示す.

送信スケジュールは, ブロードキャストフェーズと終端処理フェーズの 2 つに分割される. ブロードキャストフェーズは T 個のブロックによって構成される. 各ブロックは N 個のタイムスロットから構成され, 各ブロックの $i, i = 1, 2, \dots, N$ 番目のタイムスロットは必ず i 番目の端末に割り当てられる.

ブロック t において端末 i がブロードキャストを行う情報パケットを $\mathbf{u}_i^{(t)}$ とする. ただし, $t = 1, 2, \dots, T$ であり, $\mathbf{u}_i^{(t)}$ は i と t に関して統計的に独立である. 以降では説明の簡単化のために, スロット番号 j を以下のように定義する.

$$j = i + (t - 1)N \quad (2.1)$$

スロット番号を用いて, ブロック t において端末 i がブロードキャストする情報パケットを $\mathbf{u}_j \triangleq \mathbf{u}_i^{(t)}$ と表す.

符号化協調通信においては, 各スロット番号に対応する端末 i が自身の情報パケットをブロードキャストし, 他端末とその情報パケットを共有する. スロット番号が j のとき, それに対応する端末 i はそれまでに共有した情報パケット集合 $\{\mathbf{u}_1, \dots, \mathbf{u}_{j-1}\}$ を保持している. パリティパケット \mathbf{p}_j は, これらの情報パケットを用いて次式のように生成される.

$$\mathbf{p}_j \triangleq \odot_{l \in \mathcal{C}(j)} \mathbf{u}_l \quad (2.2)$$

ただし, パリティパケット \mathbf{p}_j のサイズは情報パケットと等しく M [bits] であり, \odot は任意の符号化操作 (例. ビットごとの排他的論理和演算) を表し, $\mathcal{C}(j)$ はスロット番号 j においてパリティパケットの生成に用いられる情報パケットのスロット集合を表す. この符号化操作 \odot と, 情報パケットのスロット集合 $\mathcal{C}(j)$ を任意に定義することによって, 符号化協調通信が定義される. 最後に, 端末 i は情報パケットとパリティパケットを接続したサイズ $2M$ [bits] の符号語をブロードキャストする.

本研究では, 符号語の変調方式として BPSK(Binary Phase-Shift Keying) を仮定する. ブロック t において端末 i が符号語を変調した送信信号ベクトルを $\mathbf{x}_i^{(t)} \in \{+1, -1\}^{2M}$ とする. このとき, 基地局における受信信号ベクトルは次式によって与えられる.

$$\mathbf{y}_i^{(t)} = h_i^{(t)} \mathbf{x}_i^{(t)} + \mathbf{n}_i^{(t)} \quad (2.3)$$

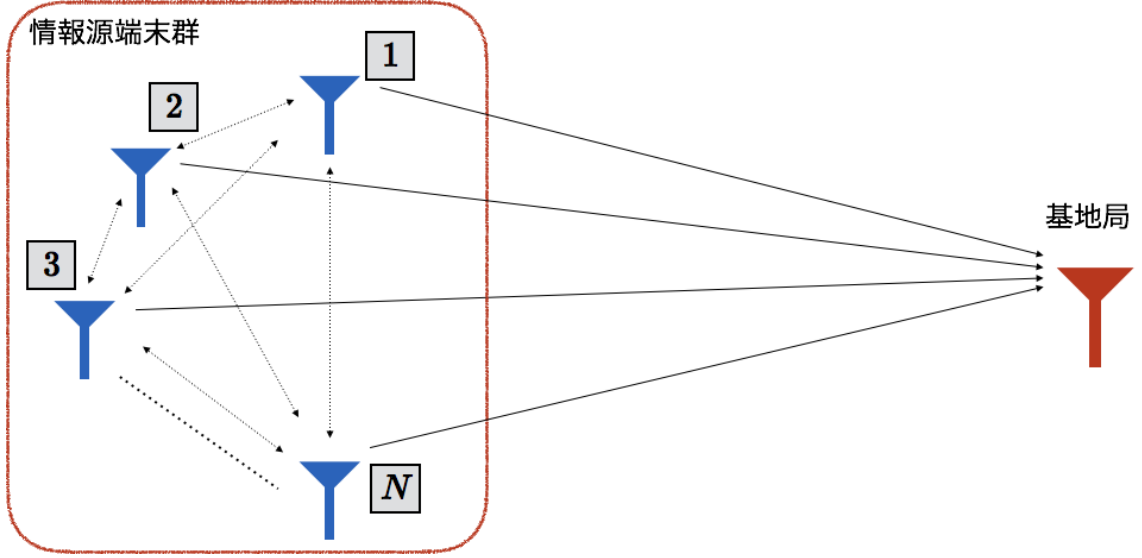


図 2.1: 本研究において想定するネットワーク

ただし、 $h_i^{(t)}$ はブロック t における端末 i と基地局間のチャネル係数であり、端末 i に関して統計的に独立である一方、ブロック t に関しては通信環境に依存する。 $h_i^{(t)}$ は平均 0, 分散 1 の円対称複素ガウス分布に従い、この値は基地局において既知とする。 $\mathbf{n}_i^{(t)}$ はサイズ $2M$ の加法的白色ガウス雑音 (AWGN: Additive White Gaussian Noise) ベクトルであり、各要素は平均 0, 分散 N_0 の円対称複素ガウス分布に従う。

各端末は端末と基地局との距離と比較して、お互いに近傍に位置していると仮定する。このとき、端末間の通信路におけるお互いの受信信号電力は十分に大きいと考えることができ、端末間の通信においては誤りは発生しないとする。

T ブロックから構成されるブロードキャストフェーズの後、 ΔT ブロックから構成される終端処理フェーズが開始される。前フェーズにおいて NT 個の全ての情報パケットが送信されているために、本フェーズにおいてはパリティパケットのみが生成され送信される。パリティパケットはスロット j におけるスロット集合 $\mathcal{C}(j)$ に対応する情報パケットを用いて、式 (2.2) により生成される。 $|\mathcal{C}(j)| = 0$ となったときに、符号化協調通信が終了する。この終端処理フェーズは、空間結合符号における畳み込み符号化の終端処理に対応する。

本研究においては、バースト誤りに対する符号化協調通信の誤り率特性を評価するために、以下の 2 つの特殊な通信路を考える。

1-ブロックレイリーフェーディング通信路

本通信路において、通信路係数は 1 ブロックごとに変化する。つまり、 $h_i^{(t)}$ は端末 i とブロック t に関して統計的に独立に変化すると言える。これは、通信路状態が緩やかに変動することに対応する。

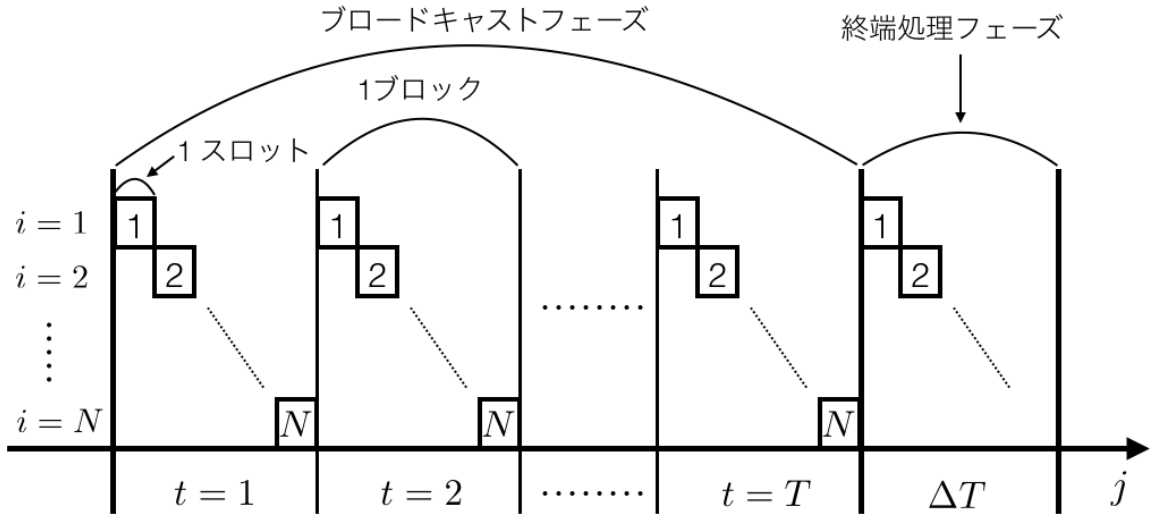


図 2.2: ネットワーク内の各端末が従う送信スケジュール

T -ブロックレイリーフェーディング通信路

本通信路において、通信路係数は $T + \Delta T$ ブロックごとに変化する。つまり、 $h_i^{(t)}$ は端末 i のみに関して統計的に独立に変化し、スロット t に関しては一切変化しない。これは、通信路状態がほとんど変動しないことに対応する。

第3章 多次元空間結合 Repeat-Accumulate 符号化協調

本章においては，空間結合 Repeat-Accumulate 符号化協調 (SC-RA-CC) と多次元空間結合 Repeat-Accumulate 符号化協調について説明する．まずはじめに，LDPC 符号の一種である RA 符号の符号化法とその復号法について説明し，空間結合 RA 符号について述べる．この符号を符号化協調通信に発展させることにより，SC-RA-CC と，MD-SC-RA-CC において，各端末が従う符号化法を説明し，符号化協調通信全体での符号化とパリティ検査行列との対応関係について述べる．さらに，パリティ検査行列から定義されるグラフ表現を用いて説明を行う．

3.1 Repeat-Accumulate 符号

本節では，RA 符号の符号化・復号方法について述べる．

3.1.1 符号化法

はじめに，RA 符号は符号化器においては，情報パケット $\mathbf{u}_i^{(t)}$ を Q 回繰り返す，サイズ QM [bits] の系列を出力する．次に出力系列の各ビットを交互に入れ替える任意のインターリーブに入力にする．そして，ベクトルの先頭から a [bits] ずつの要素の排他的論理和をとる．最後に Accumulator と言われる伝達関数 $1/(1-D)$ の畳み込み符号で符号化することにより所望の符号語が得られる．以降において，上記の符号化を詳しく説明し，その検査行列表現，グラフ表現を説明する．

(1) 符号化器

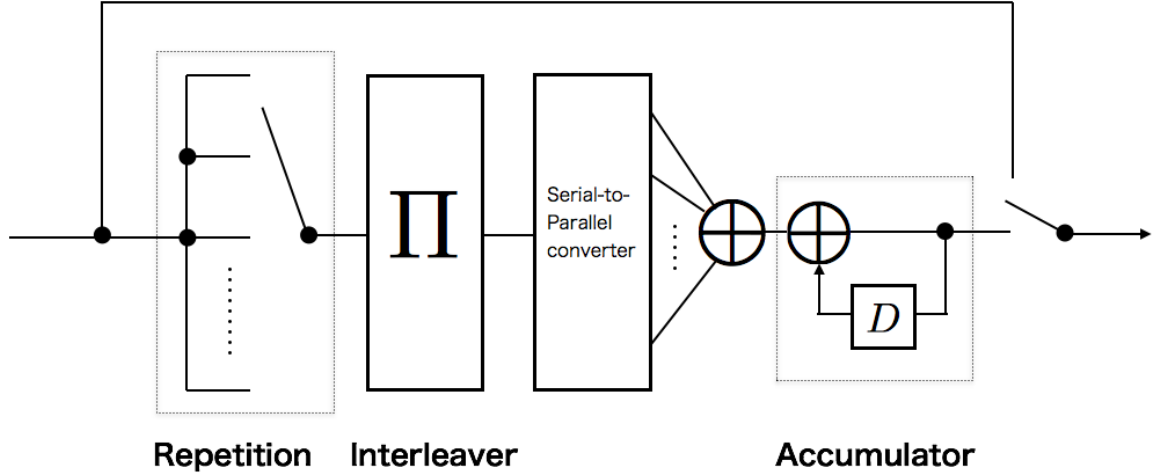
RA 符号化器を図 3.1 に示し， \mathcal{D} は 1bit 遅延器 (Delay Device)， \oplus は排他的論理和演算子を表す．

まず情報パケットを構成する各要素 $\mathbf{u} = [u_1, \dots, u_m, \dots, u_M]$ は Repetition によって Q 回繰り返され，サイズ QM の \mathbf{u}' が生成される．ただし， u_m はベクトル \mathbf{u} の第 m 番目の要素を表す．この生成された出力系列は次式で表され，そのサイズは QM [bits] となる．

$$\mathbf{u}' = [u_1, \dots, u_1, \dots, u_M, \dots, u_M] \quad (3.1)$$

次に，出力系列 \mathbf{u}' の各要素は Interleaver 行列 H によって並び替えられる．行列 H は行重みと列重みがともに 1 である 2 元 $QM \times QM$ 行列である．インターリーブ後のベクトル \mathbf{w} は以下の式で表される．

$$\mathbf{w} = \mathbf{u}'H \quad (3.2)$$


 図 3.1: RA 符号化器 (D は 1bit 遅延器 (Delay Device), \oplus は排他的論理和演算子)

そして、インターリーブされた出力系列 w は先頭の要素から順番に a bits ずつ排他的論理和演算子に入力され、サイズ $L = (Q/a)M$ の出力系列 s が生成され、 s は次式で表される。

$$\begin{aligned} \mathbf{s} &= [w_1 \oplus \cdots \oplus w_a, \dots, w_{(QM-a+1)} \oplus \cdots \oplus w_{(QM)}] \\ &= [s_1, \dots, s_L] \end{aligned} \quad (3.3)$$

この処理は Combiner と呼ばれ、 a bits の排他的論理和を計算する Combiner を、レート a の Combiner として定義する。

Combiner からの出力系列 s を Accumulator と呼ばれる畳み込み符号化器に入力し、パリティベクトル p を生成する。Accumulator は図 3.1 に示すように 1 つの 1 ビット遅延器と 1 つの排他的論理演算子から構成されるフィードバック回路であり、この畳み込み符号の伝達関数は $1/(1-D)$ で表される。所望のパリティパケットはそのサイズを L bits とし、次式で定義される。

$$\begin{aligned} \mathbf{p} &= [s_1, s_2 \oplus p_1, \dots, s_L \oplus p_{(L-1)}] \\ &= [p_1, \dots, p_L] \end{aligned} \quad (3.4)$$

c 最後に、本節では組織符号化の RA 符号を考えるため、情報パケット u とパリティパケット p を結合し、次式で示す符号語ベクトル c を得る。

$$\mathbf{c} = \begin{bmatrix} \mathbf{u} & \mathbf{p} \end{bmatrix} \quad (3.5)$$

このとき (Q, a) -regular RA 符号の符号化率 r_{RA} は以下の式で表される。

$$r_{RA} = \frac{M}{M+L} = \frac{M}{\left(1 + \frac{Q}{a}\right)M} = \frac{a}{a+Q} \quad (3.6)$$

つまり、RA 符号の符号化率は Repetition の繰り返し回数 Q と Combiner の a のみに依存する。

(2) 検査行列表現

本節では、前節で説明した RA 符号化器の構造から定義される検査行列を説明し、符号化器との関係性について述べる。

(Q, a) -regular RA 符号を定義する検査行列 \mathbf{H}_{RA} は $N \times (M + L)$ 行列であり、 $L \times M$ 行列 B と $L \times N$ 行列 A を用いて、次式のように表すことができる。

$$\mathbf{H}_{RA} = \begin{bmatrix} B & A \end{bmatrix} \quad (3.7)$$

行列 B は符号化器のベクトル \mathbf{u}' , \mathbf{w} , \mathbf{s} の生成処理に対応し、列重み Q , 行重み a の条件のもと 0 と 1 がランダムに配置された行列である。行列 B は、転置行列 B^T に左から情報パケット系列 \mathbf{u} を乗算することにより、ベクトル \mathbf{s} を得られたことから、符号化器の構造に対応することがわかる。また、行列 B の行重みはパラメータ Q , 列重みはパラメータ a によって決定される。

次に行列 A は $L \times L$ 行列であり、以下の式によって定義される。

$$A = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & \ddots & & \\ & & & \ddots & 1 & \\ & & & & 1 & 1 \\ & & & & & 1 & 1 \end{bmatrix} \quad (3.8)$$

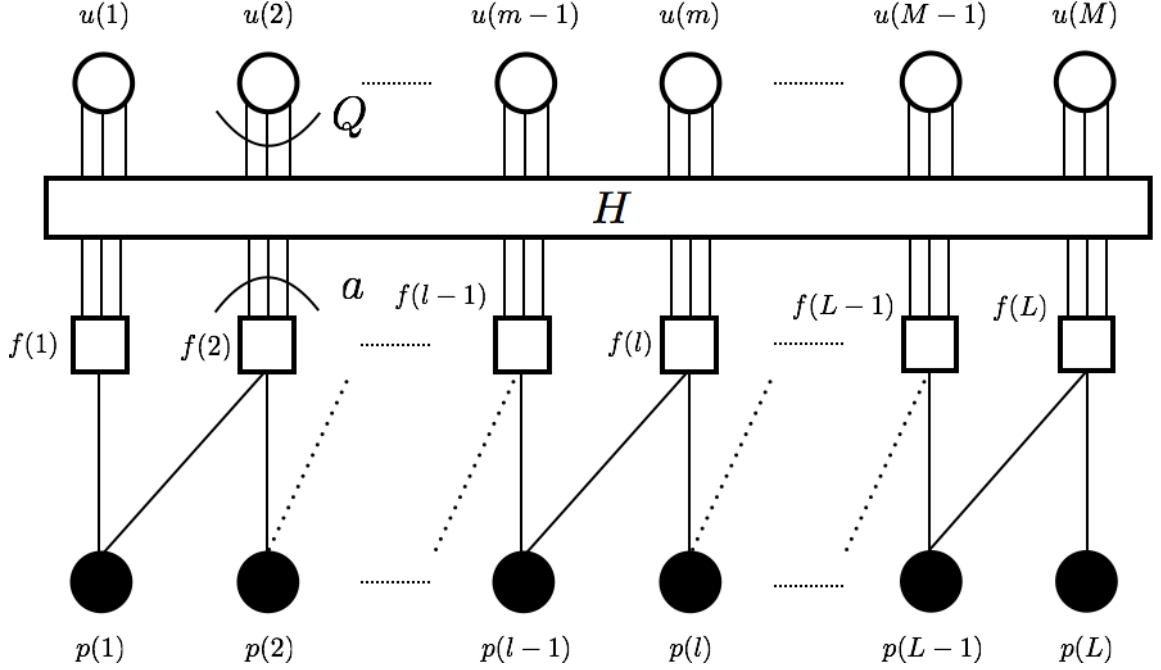
これは、行重みと列重みがともに 2 (ただし 1 行目と M 列目の重みは 1) の帯対角行列であり、Accumulator の処理の行列表現である。この行列 A の特徴について説明する。まず検査行列 \mathbf{H}_{RA} に対して行基本変形を行う。第 $l \in \{2, \dots, L\}$ 番目の行に第 $l-1$ 番目の行を加えると、行列 A の帯対角行列は対角行列となり、行列 B は式 (3.4) の操作を行った行列と等価になる。よって、行列 A が符号化器における Accumulator の処理を表すと言える。以上より、サイズ $L \times (M + L)$ の検査行列 \mathbf{H}_{RA} は RA 符号の符号化器から定義され、符号語ベクトル \mathbf{c} に対して、次式が成立する。

$$\mathbf{H}_{RA} \mathbf{c}^T = 0 \quad (3.9)$$

3.1.2 グラフ表現

本節では、前節の RA 符号の検査行列により定義される無向性 2 部グラフであるタナーグラフを説明する。図 3.2 に検査行列 \mathbf{H}_{RA} によって定義されるタナーグラフを示す。タナーグラフを構成するノードは 3 つのグループに分けられる。1 つ目のグループは情報パケット系列 \mathbf{u} の M 個の要素に対応し M 個のノードからなるメッセージノード (白丸 \circ)、2 つ目のグループはベクトル \mathbf{s} の N 個の要素に対応し、パリティパケットの制約条件を与える L 個のノードからなるチェックノード (四角 (\square))、3 つ目はパリティパケット系列 \mathbf{p} の L 個の要素に対応し L 個のノードからなるパリティノード (黒丸 \bullet) である。

以降、 m 番目のメッセージベクトル要素に対応するメッセージノードを $u(m)$ と表記する。チェックノード、パリティノードについても同様に、 $f(l)$, $p(l)$, $l \in \{1, \dots, L\}$ と表記する。


 図 3.2: (Q, a) -regular RA 符号のタナーグラフ表現

ここで、メッセージノードには $u(m)$ $m \in \{1, \dots, M\}$, チェックノードは $f(l)$ $l \in \{1, \dots, L\}$, パリティノードは $p(l)$ $l \in \{1, \dots, L\}$ の行列番号に対応した固有の記号がそれぞれ与えられる. 検査行列 \mathbf{H}_{RA} を構成する行列 B の l 行 m 列目要素が 1 ならば, タナーグラフにおいてメッセージノード $u(m)$ とチェックノード $f(l)$ がエッジで結ばれる. パリティノードとチェックノードのエッジの結合も同様である. このようにして定義されたタナーグラフは, あるノードに連結しているエッジの数とそのノードに対応する検査行列上の行 (もしくは列) の重みは等しくなる. Interleaver 行列 H はメッセージノードとチェックノードの間のエッジの結合を決定し, すべてのメッセージノードは Q 本のエッジを持つ. 一方, 各チェックノードのメッセージノード側に出るエッジの数はすべて a 本であり, このようなタナーグラフの構造を持つ RA 符号を (Q, a) -regular RA 符号と呼ばれる. つまり, 各メッセージノードとチェックノード間において入力されるエッジの数が全て等しいときに, この RA 符号は Regular と呼ばれる. 一方, この定義とは反対に各メッセージノードとチェックノード間で結合されるエッジの数が等しくなくある確率密度分布を持つ確率変数に従うとき, この RA 符号は Irregular RA 符号と定義される.

まず, メッセージノード $u(m)$ はベクトル \mathbf{u} の要素 u_m に対応しており, メッセージノード $u(m)$ から出る Q 本のエッジによってベクトル \mathbf{u} の要素 u_m が Q 回繰り返される. そして Interleaving 行列 H によりメッセージノード $u(m)$ の持つエッジはチェックノード $f(l') \in D(u(m))$ に結合される. ただし, $D(u(m))$ はメッセージノード $u(m)$ と結合するチェックノードの集合であり, 要素数は Q である.

次に, チェックノード $f(l)$ の持つエッジはメッセージノード $u(m') \in E(f(l)) \setminus \{p(l), p(l-1)\}$ と結合される. ただし, $E(f(l))$ はチェックノード $f(l)$ と結合するメッセージノードとパリティノード

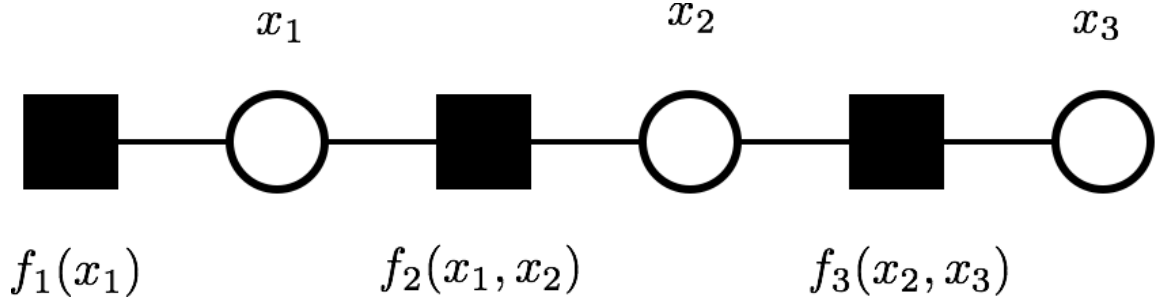


図 3.3: 3 変数ファクターグラフの一例

ドの集合で要素数は $a + 2$ であり，記号 \setminus は左の集合に右の要素を含まないことを示す．ただし， $l = 1$ のとき要素数は $a + 1$ である．この操作はチェックノード $f(l)$ と結合するメッセージノードに対応するベクトル要素 $u_{m'}$ の排他的論理和をとることであり，式 (3.2) と一致する．

そして，パリティノード $p(l)$ がチェックノード $f(l)$ ， $f(l + 1)$ とエッジで結合している構造は Accumulator の処理である式 (3.4) に対応する．よって，タナグラフ上のパリティノードは符号化されたパリティベクトル \mathbf{p} と一致する．以上から，タナグラフは検査行列によって定義され RA 符号化器の構造を表すことができる．

3.1.3 復号法

本節では，本章で説明してきた RA 符号の復号法 [23, 24] を説明する．まず多変数関数の周辺化問題を定義し，その周辺化を高速に行う sum-product アルゴリズムを導入する．そして，与えられた観測値に基づく確率推論に sum-product アルゴリズムを適用できることを示し，RA 符号の復号に利用される sum-product 復号法，特に対数領域 sum-product 復号法を示す [25]．ただし，本節で用いる記号 $f(x)$ と $D(v)$ は前節のものと関係を持たない．

(1) 多変数関数の周辺化問題

集合 $V \triangleq \{x_1, \dots, x_I\}$ を I 個の変数の集合とし， x_i ， $i \in \{1, \dots, I\}$ のとりうる値の集合を $D(x_i)$ とする．ここで，変数集合 V に依存する多変数関数 $f(V) = f(x_1, \dots, x_I)$ を考えると，変数 x_i に関する周辺化関数は以下の式で表される．

$$g_i(x_i) \triangleq \sum_{V \setminus x_i} f(V) \quad (3.10)$$

表記 $V \setminus x_i$ は， x_i を除く V に含まれるすべての変数がすべての可能な値 (定義域内のすべての値) をとることを意味し，関数 $f(V)$ をグローバル関数と呼ばれる．周辺化問題は上記の周辺化関数を求める問題と定式化される．

しかし，周辺化問題は一般に計算量的に困難な問題となる．その理由について例を用いて説明する．簡単のために， $D(x_i) = \{0, 1\}$ であるすべての変数の定義域が 2 値である場合について考

える．周辺化関数の式 (3.10) を定義式に従って計算する場合には， $I - 1$ 個の変数についての和をとる必要があり，そのパターン数は 2^{I-1} となる．定義式 (3.10) 通りに行うと，この和の評価には変数の個数 I に対して指数的な演算量が必要になる．よって，周辺化問題を解くことは計算量的に困難であると言える．

I 個の変数を含むグローバル関数 $f(V)$ の因子分解が以下の式で与えられているものとする．

$$f(V) = f_1(A_1) \dots f_J(A_J) \quad (3.11)$$

A_1, \dots, A_J は変数集合 V の部分集合であり，それぞれの因子 $f_j(A_j)$ ， $j \in \{1, \dots, J\}$ をローカル関数と呼ぶ．ここで，多変数関数 $f(V)$ の因子分解を視覚的に示すグラフ表現として，ファクターグラフを導入する．上記の因子分解に対応するファクターグラフは，(1) ローカル関数 $f_j(A_j)$ に対応する因子ノード，(2) 変数 $x_i \in V$ に対応する変数ノードの 2 種のノードから構成される． $f_j(A_j)$ に対応する因子ノードと A_j に含まれる変数に対応する変数ノード間にはエッジが存在し，エッジが存在するのはこの条件が成り立つときに限られる．

一つの例として，因子分解

$$f(x_1, x_2, x_3) = f_1(x_1)f_2(x_1, x_2)f_3(x_2, x_3) \quad (3.12)$$

に対応するファクターグラフを図 3.3 に示す．ただし， $V = \{x_1, x_2, x_3\}$ ， $A_1 = \{x_1\}$ ， $A_2 = \{x_1, x_2\}$ ， $A_3 = \{x_2, x_3\}$ であり，白丸 (○) は変数ノード，黒四角 (■) は因子ノードにそれぞれ一致する．これはファクターグラフの定義通り，因子ノード $f_2(x_1, x_2)$ の場合，変数ノード x_1 と x_2 と結合していることがわかる．

(2) sum-product アルゴリズム

sum-product アルゴリズムは，前節で説明した多変数関数の周辺化問題を高速に解くためのアルゴリズムであり，ファクターグラフ上でのメッセージ交換処理に基づくアルゴリズムである．グローバル関数 $f(V)$ が多数のローカル関数 $f_j(A_j)$ の積に因子分解できる場合には，その関数の構造を利用し周辺化問題を解くための計算量を大幅に削減できる．

まず，関数周辺化問題に現れる計算は，ローカル関数 $f_j(A_j)$ の積の和であり，積和計算の形をしている．そこで，積和計算の演算量を低減するために分配則が利用される．与えられた積和を計算量の少ない形に変換するには，変数の依存関係に注意して和記号の位置と括弧の位置 (演算の順番) を動かすことが必要である．分配則を利用した周辺化計算の一例を示す．

関数 $f(x_1, x_2, x_3, x_4)$ に対して，因子分解

$$f(x_1, x_2, x_3, x_4) = f_1(x_1)f_2(x_1, x_2)f_3(x_1, x_3)f_4(x_3, x_4) \quad (3.13)$$

が与えられている．周辺化関数 $g_1(x_1)$ を求めるためには，

$$g_1(x_1) = \sum_{x_2} \sum_{x_3} \sum_{x_4} f_1(x_1)f_2(x_1, x_2)f_3(x_1, x_3)f_4(x_3, x_4) \quad (3.14)$$

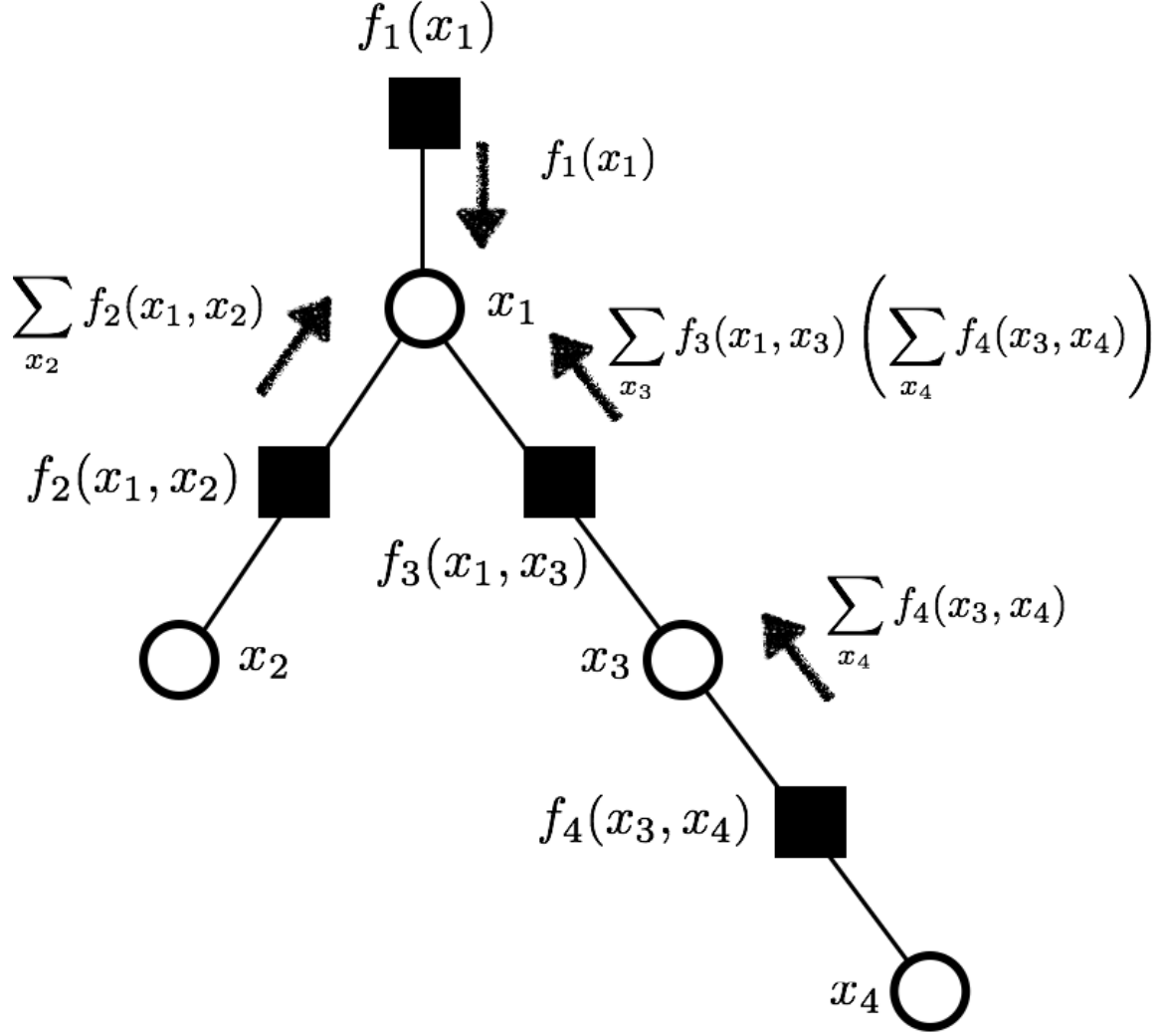


図 3.4: メッセージ交換による周辺化計算の一例

を評価する必要があるが、この式を次式のように和記号の位置と括弧の位置を動かすことにより、計算量を小さくすることができる。

$$g_1(x_1) = f_1(x_1) \left(\sum_{x_2} f_2(x_1, x_2) \right) \left(\sum_{x_3} f_3(x_1, x_3) \left(\sum_{x_4} f_4(x_3, x_4) \right) \right) \quad (3.15)$$

$|D(x_i)| = 10$, $i \in \{1, \dots, 4\}$ と仮定した場合、式 (3.14) では $10^3 - 1 = 999$ 回の和を評価する必要がある一方、分配則を利用した式 (3.15) では 108 回の和で関数の周辺化が行われる。式 (3.15) の計算はグローバル関数 $f(x_1, x_2, x_3, x_4)$ のファクターグラフ上で局所積和を伝搬させているものと見ることができる。図 3.4 のように変数ノード x_1 では、そのノードに流れ込むすべての局所積和をとることにより、周辺化関数 $g_1(x_1)$ が得られており、式 (3.15) の計算と一致する。

sum-product アルゴリズムは、先に述べた分配則による演算量削減を系統的に行う手法であり、その処理は以下の流れに従う。

ステップ 1 (ファクターグラフの構成) 対象とするグローバル関数 $f(V)$ の因子分解が、式 (3.11) によって与えられているとする。この因子分解に対応するファクターグラフを G とする。以下では、ファクターグラフは連結グラフであり、かつループを持たないと仮定する。また、変数 x_i に対応する変数ノードの名前を x_i とし、因子 f_j に対応する因子ノードの名前を f_j とする。

計算したい周辺化関数を $g_r(x_r)$, $r \in \{1, \dots, N\}$ とするとき、 G を x_r をルートとする木とみなし、グラフ G はルートを上にして描かれているものとする。最後に、ファクターグラフ G に含まれるノード v に結合するノードの集合を $N(v)$ とする。

ステップ 2 (変数ノード処理) 変数ノード x_k から因子ノード f_i へのメッセージは以下のように計算される。

$$M_{x_k \rightarrow f_i}(x_k) = \prod_{a \in N(x_k) \setminus f_i} M_{a \rightarrow x_k}(x_k) \quad (3.16)$$

ただし、 x_k が葉ノードの場合には、初期条件として $M_{x_k \rightarrow f_i}(x_k) = 1$ とする。この記法は $M_{x_k \rightarrow f_i}(x_k)$ は、 $x_k \rightarrow f_i$ 方向に伝達されるメッセージであり、変数 x_k の関数になっていることを示す。

ステップ 3 (因子ノード処理) 因子ノード f_i から変数ノード x_k へのメッセージは以下のように計算される。

$$M_{f_i \rightarrow x_k}(x_k) = \sum_{A_i \in N(f_i) \setminus x_k} f_i(A_i) \prod_{a \in N(f_i) \setminus x_k} M_{a \rightarrow f_i}(a) \quad (3.17)$$

ただし、 f_i が葉ノードの場合、初期条件として $M_{f_i \rightarrow x_k}(x_k) = f_i(x_k)$ とする。

ステップ 4 (周辺化関数の計算) ステップ 2 とステップ 3 をファクターグラフの下流から上流に向けて順に計算を進める。ルートノード x_r にメッセージが到達したとき、ルートノード x_r において以下の計算を行う。

$$M_{x_r}(x_r) = \prod_{a \in N(x_r)} M_{a \rightarrow x_r}(x_r) \quad (3.18)$$

すなわち、ルートノード x_r は下流から到達したすべてのメッセージの積を計算することであり、この積が周辺化関数 $g_r(x_r)$ と一致する。

ここまでは、一つの変数に対する周辺化関数の計算方法を示してきたが、すべての変数についてその周辺化関数を効率良くおこなう計算方法について述べる。その手法は同時更新 sum-product アルゴリズムと呼ばれ、以下の手順によって行われる。

1. 変数ノードから因子ノードの方向にメッセージを送る。
2. すべての因子ノードにおいて因子ノードから変数ノード方向へのメッセージを計算し、結合する変数ノードへ送る。
3. すべての変数ノードにおいて変数ノードから因子ノード方向へのメッセージを計算し、結合する因子ノードへ送る。

4. メッセージが収束したかどうかを判定し、収束していなければ、"2"に戻る。
5. ローカル関数に相当する積を各ノードにおいて計算する。

上記に示した同時更新 sum-product アルゴリズムは、因子ノードの処理とメッセージ送信、変数ノード処理とメッセージ送信が交互に一斉に行われる。しかし、更新のスケジュールには様々なものが存在し、ファクターグラフの構造により望ましいスケジュールは異なる。

(3) シンボル MAP 復号法

本節では、シンボル MAP(Maximum A posteriori Probability) 復号法の定義を述べ、前節で説明した周辺化問題との関係性について述べる。ここでは、送信アルファベットが 2 値 $\chi = \{0, 1\}$ であるとし、受信アルファベットを \mathcal{Y} とする。また、条件付確率 $P_{Y|X}(\mathbf{y}|\mathbf{x})$ により定義される通信路を仮定し、送信語は符号長 N の 2 元線形符号 \mathcal{C} の符号語であるとする。受信器が $\mathbf{y} \in \mathcal{Y}^N$ を受信したという仮定でベイズ則に基づく計算を行うと、以下の式で定義される事後確率が得られる。

$$P_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{P_X(\mathbf{x})P_{Y|X}(\mathbf{y}|\mathbf{x})}{P_Y(\mathbf{y})} \quad (3.19)$$

送信符号語を表す確率変数を $X = (X_1, \dots, X_N)$ とするとき、上記の事後確率は次式のように書き直すことができる。

$$P_{X|Y}(\mathbf{x}|\mathbf{y}) = P_{X_1 \dots X_N|Y}(x_1, \dots, x_N|\mathbf{y}) \quad (3.20)$$

ここで、変数集合 V を $V \triangleq \{x_1, \dots, x_N\}$ と定義し、ある送信シンボル X_n , $n \in \{1, \dots, N\}$ に注目すると、その事後確率は次式で示される。

$$P_{X_n|Y}(x_n|\mathbf{y}) = \sum_{V \setminus x_n} P_{X_1 \dots X_N|Y}(x_1, \dots, x_N|\mathbf{y}) \quad (3.21)$$

この $P_{X_n|Y}(x_n|\mathbf{y})$ は、事後確率を周辺化して得られるため、周辺事後確率と呼ばれる。ここで、 $x_n = 0$ の時の事後確率は、次式のように表現され、

$$P_{X_n|Y}(0|\mathbf{y}) = \frac{1}{P_Y(\mathbf{y})} \sum_{V \setminus x_n} P_{X_1 \dots X_N}(x_1, \dots, x_{n-1}, 0, x_{n+1}, \dots, x_N) \times P_{Y|X_1 \dots X_N}(\mathbf{y}|x_1, \dots, x_{n-1}, 0, x_{n+1}, \dots, x_N) \quad (3.22)$$

2 元線形符号 \mathcal{C} の符号語以外を送信する確率は 0、つまり $P_X(\mathbf{x}) = 0$ ($\mathbf{x} \notin \mathcal{C}$) であるので、上式の事前確率は次のように簡単化される。

$$P_{X_n|Y}(0|\mathbf{y}) = \frac{1}{P_Y(\mathbf{y})} \sum_{\mathbf{x} \in \mathcal{C}, x_n=0} P_X(\mathbf{x})P_{Y|X}(\mathbf{y}|\mathbf{x}) \quad (3.23)$$

同様に、 $x_n = 1$ のときは次式で与えられる。

$$P_{X_n|Y}(1|\mathbf{y}) = \frac{1}{P_Y(\mathbf{y})} \sum_{\mathbf{x} \in \mathcal{C}, x_n=1} P_X(\mathbf{x})P_{Y|X}(\mathbf{y}|\mathbf{x}) \quad (3.24)$$

以上から、シンボル MAP 復号法によって、第 n 送信シンボルの推定値 \hat{x}_n は式 (3.23) と式 (3.24) で得られたシンボル単位の事後確率に基づいて決定される。すべてのシンボル x_n に対して対数事後確率比

$$\begin{aligned} L_n &= \ln \frac{P_{X_n|Y}(0|\mathbf{y})}{P_{X_n|Y}(1|\mathbf{y})} \\ &= \ln \frac{\sum_{\mathbf{x} \in \mathcal{C}, x_n=0} P_X(\mathbf{x}) P_{Y|X}(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{x} \in \mathcal{C}, x_n=1} P_X(\mathbf{x}) P_{Y|X}(\mathbf{y}|\mathbf{x})} \end{aligned} \quad (3.25)$$

を計算した上で推定シンボル \hat{x}_n を次のように定める。

$$\hat{x}_n = \begin{cases} 0 & L_n \geq 0 \\ 1 & L_n < 0 \end{cases} \quad (3.26)$$

式 (3.23) と式 (3.24) の値を求めるためには、式中の多変数関数の周辺化を行う必要があり、この計算は前節で説明した sum-product アルゴリズムを用いることにより高速に解を求めることが可能である。よって、sum-product アルゴリズムは与えられた観測値に基づく確率推論に適用できることがわかる。

(4) 対数領域 sum-product 復号法

対数領域 sum-product 復号法は、すべてのメッセージを対数比の形で表現し、対数領域においてメッセージの更新を行うアルゴリズムである。RA 符号のような 2 元低密度パリティ検査 (LDPC : Low Density Parity Check) 符号の場合、アルゴリズムが簡潔になり数値計算的にも安定するため、ソフトウェア実装、ハードウェア実装に向いている。対数領域 sum-product 復号法の詳細を以下に示す。RA 符号の復号器が受信信号ベクトル $\mathbf{y} \in \mathbb{R}^{M+L}$ を受け取ったとする。これは第 2 章の式 (2.3) により定義される。この受信信号ベクトルを用いて以下の復号法を実行し、符号語を推定する。

ステップ 1 (初期化) 各チェックノード $f(l)$ の持つすべてのエッジに対して、対数外部値比 $\alpha_{f(l) \rightarrow \psi_\alpha}$, $\psi_\alpha \in E(f(l))$ を以下の式で定義し 0 を代入する。次に、各メッセージノード $u(m)$ の持つすべてのエッジに対して、対数事前値比 $\beta_{u(m) \rightarrow \psi}$, $\psi \in D(u(m))$ を以下の式で定義し 0 を代入する。最後に、各パリティノード $p(l)$ の持つ 2 つのエッジに対して、対数事前値比 $\beta_{p(l) \rightarrow \psi_\beta}$, $\psi_\beta \in \{f(l), f(l+1)\}$ を定義し 0 を代入する。ただし、 ψ はノード間のエッジの集合の一つの要素を表す。

$$\alpha_{f(l) \rightarrow \psi_\alpha} \triangleq \ln \left(\frac{M_{f(l) \rightarrow \psi_\alpha}(0)}{M_{f(l) \rightarrow \psi_\alpha}(1)} \right) \quad (3.27)$$

$$\beta_{p(l) \rightarrow \psi_\beta} \triangleq \ln \left(\frac{M_{p(l) \rightarrow \psi_\beta}(0)}{M_{p(l) \rightarrow \psi_\beta}(1)} \right) \quad (3.28)$$

また、確率伝搬の反復回数を表す変数を $It = 0$ とし、反復最大回数を変数 It_{\max} に設定する。

ステップ 2 (対数尤度比の計算) AWGN 通信路における受信信号ベクトル \mathbf{y} の第 j 番目の要素 y_j を含む尤度関数 $P(y_j|x_j)$ は次式で表される.

$$P(y_j|x_j) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_j - x_j)^2}{2\sigma^2}\right) \quad (3.29)$$

このとき, y_j に関する対数尤度比 λ_j は次式で表される.

$$\begin{aligned} \lambda_j &\triangleq \ln \frac{P(y_j|x_j = 1)}{P(y_j|x_j = -1)} \\ &= \frac{-(y_j - 1)^2 + (y_j + 1)^2}{2\sigma^2} \\ &= \frac{2y_j}{\sigma^2} \end{aligned} \quad (3.30)$$

ただし, x_j は送信信号ベクトル \mathbf{x} の第 j 番目の要素であり, $j \in \{1, \dots, M+L\}$ である. ここから, 2 元入力 AWGN 通信路の場合の対数尤度比 λ_j は受信信号 y_j と通信路の雑音 σ^2 のみにより決定されることがわかる. 全ての要素 $j \in \{1, \dots, M+L\}$ に対して式 (3.30) を適用し, 対数尤度比ベクトル $\Lambda \in \mathbb{R}^{M+L}$ が得られる. また, 受信信号ベクトル \mathbf{y} の第 j 番目の要素 y_j がパンクチャリングされているとき対数尤度比 λ_j は 0 となる.

ステップ 3 (パリティノード処理) パリティノード $p(l)$, $l \in \{1, \dots, L\}$ に対して以下の計算を行い, 対数事前値比 β を更新する.

$$\beta_{p(l) \rightarrow f(l')} = \begin{cases} \lambda_{p(l)} & l = L \text{ のとき} \\ \lambda_{p(l)} + \alpha_{f(l'') \rightarrow p(l)} & \text{それ以外のとき} \end{cases} \quad (3.31)$$

ただし, $(f(l'), f(l'')) = (f(l), f(l+1)), (f(l+1), f(l))$ である.

ステップ 4 (メッセージノード処理) メッセージノード $u(m)$, $m \in \{1, \dots, M\}$ に対して以下の計算を行い, 対数事前値比 β を更新する.

$$\beta_{u(m) \rightarrow f(l)} = \lambda_{u(m)} + \sum_{f(l') \in D(u(m)) \setminus f(l)} \alpha_{f(l') \rightarrow u(m)} \quad (3.32)$$

ステップ 5 (チェックノード処理) チェックノード $f(l)$, $l \in \{1, \dots, L\}$ に対して以下の計算を行い, 対数外部値比 α を更新する.

$$\alpha_{f(l) \rightarrow p(l)} = \left(\prod_{\psi \in E(f(l)) \setminus p(l)} \text{sign}(\beta_{\psi \rightarrow f(l)}) \right) f \left(\sum_{\psi \in E(f(l)) \setminus p(l)} f(|\beta_{\psi \rightarrow f(l)}|) \right) \quad (3.33)$$

$$\alpha_{f(l) \rightarrow u(m)} = \left(\prod_{\psi \in E(f(l)) \setminus u(m)} \text{sign}(\beta_{\psi \rightarrow f(l)}) \right) f \left(\sum_{\psi \in E(f(l)) \setminus u(m)} f(|\beta_{\psi \rightarrow f(l)}|) \right) \quad (3.34)$$

ただし,

$$\text{sign}(x) \triangleq \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \quad (3.35)$$

であり, 関数 $f(x)$ は

$$f(x) \triangleq \ln \frac{\exp(x) + 1}{\exp(x) - 1} \quad (3.36)$$

と定義される.

ステップ 6 (一時推定語の導出) メッセージベクトル \mathbf{u} の第 m 番目の要素 u_m の一時推定語 \hat{u}_m は以下の式で表され,

$$\hat{u}_m = \begin{cases} 0 & \text{sign} \left(\sum_{f(l') \in D(u(m))} \beta_{u(m) \rightarrow f(l')} \right) = 1 \text{ のとき} \\ 1 & \text{sign} \left(\sum_{f(l') \in D(u(m))} \beta_{u(m) \rightarrow f(l')} \right) = -1 \text{ のとき} \end{cases} \quad (3.37)$$

また, パリティベクトル \mathbf{p} の第 l 番目の要素 p_l の一時推定語 \hat{p}_l は以下の式で表される.

$$\hat{p}_l = \begin{cases} 0 & \text{sign} \left(\sum_{f(l') \in \{f(l), f(l+1)\}} \beta_{p(l) \rightarrow f(l')} \right) = 1 \text{ のとき} \\ 1 & \text{sign} \left(\sum_{f(l') \in \{f(l), f(l+1)\}} \beta_{p(l) \rightarrow f(l')} \right) = -1 \text{ のとき} \end{cases} \quad (3.38)$$

ただし, $l = L$ のときパリティノード $p(l)$ に結合するチェックノードは一つだけなので $f(l') \in \{f(l)\}$ となる. 以上から, 一時推定語ベクトル $\hat{\mathbf{c}}$ を以下のように定義する.

$$\hat{\mathbf{c}} = \begin{bmatrix} \hat{u}_1 & \dots & \hat{u}_M & \hat{p}_1 & \dots & \hat{p}_L \end{bmatrix} \quad (3.39)$$

ステップ 7 (パリティ検査) 一時推定語ベクトル $\hat{\mathbf{c}}$ が誤りなく符号語ベクトル \mathbf{c} に復号されているかを検査する. もし, 一時推定語ベクトル $\hat{\mathbf{c}}$ が

$$\hat{\mathbf{c}} \mathbf{H}_{RA}^T = 0 \quad (3.40)$$

を満たせば, ベクトル $\hat{\mathbf{c}}$ を推定語ベクトルとして出力し, アルゴリズムを終了する.

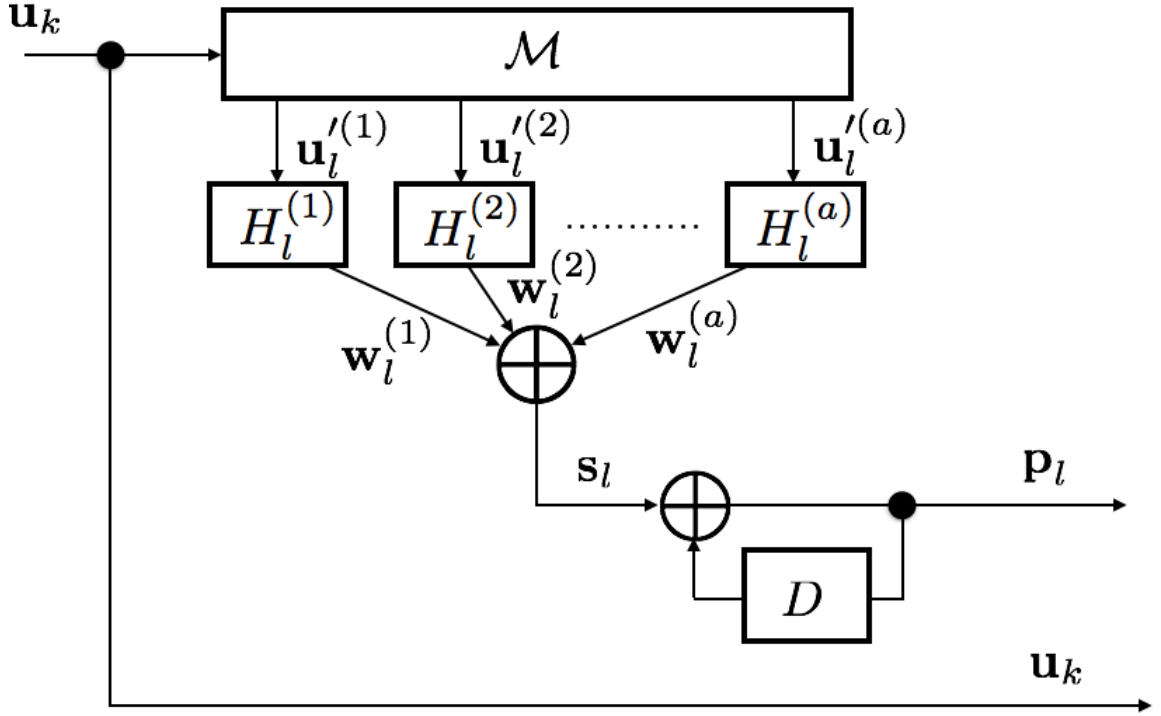
ステップ 8 (反復回数の更新) 反復回数の変数 It を 1 だけ増加させ, もし $It < It_{\max}$ ならばステップ 3 に戻る. それ以外のときは, 一時推定語ベクトル $\hat{\mathbf{c}}$ を推定語ベクトルとして出力し, アルゴリズムを終了する.

3.2 空間結合 RA 符号

本節では, 空間結合 RA 符号 [14] を説明し, その符号のグラフ表現について述べる.

3.2.1 符号化法

本節では, 空間結合 RA 符号の符号化器に基づく符号化法を説明する. また, 符号化器に対応した検査行列表現とグラフ表現を導入しその関係性について述べる.


 図 3.5: (Q, a, L) -空間結合 RA 符号の符号化器の構造

(1) 符号化器

空間結合 RA 符号の符号化器の構造を図 3.5 に示す。図中の記号 \mathcal{M} は 2 元有限体 \mathbb{F}_2 を要素にもちサイズ M のベクトルを格納するメモリを意味し、格納できるベクトルの数は任意である。

まず初めに、符号化器に情報パケット $\mathbf{u}_k \in \mathbb{F}_2^M$ が入力される。符号化器に入力された情報パケット \mathbf{u}_k はメモリ \mathcal{M} に格納される。メモリに格納されているメッセージベクトル $\mathbf{u}_{k'}, k' \in \{1, \dots, k\}$ はそれぞれ計 Q 回かつ、ベクトル番号 k' が小さいものから順にという条件のもと、メモリから a 個のベクトルが出力される。ただし、メモリ内の残存ベクトル数が a 未満の場合は、メモリ内の全てのベクトルが出力される。ただし、メモリから Q 回出力されたメッセージベクトル $\mathbf{u}_{k'}$ はメモリから消去される。また、ベクトル $\mathbf{u}_l^{(a')}$ の右下の添字 $l \in \{1, \dots, L\}$ は符号化器で生成されるパリティベクトルの番号であり、計 L 個のパリティベクトルが生成される。 $l \in \mathbb{N}$ であることから、 l が 1 から $a-1$ のとき出力されるベクトルの数は a 未満となり、また l が $L-a+1$ から L のときも同様に a 未満となる。これは、空間結合 RA 符号の符号化器がメモリ \mathcal{M} を所持していることの特徴である。

次に、Interleaving についてである。ベクトル $\mathbf{u}_l^{(a')}$ はそれぞれの番号 a' に対応する Interleaver に入力され、その処理は Interleaving 行列 $H_l^{(a')}$ によって決定される。ただし、入力ベクトルが存在しないとき Interleaving 行列は存在しないものとする。行列 $H_l^{(a')}$ は 2 元 $M \times M$ 行列であり、行重みと列重みがともに 1 となる制約のもとランダムに 1 と 0 が配置された行列である。第 (1) 節で紹介した行列 H とは行列のサイズと重みが異なることに注意したい。またすべての行列 $H_l^{(a')}$ は a' や l の値に関係なく、統計的に独立である。この Interleaving 行列 $H_l^{(a')}$ と第 (1) 節の

式 (3.2) によって、入れ替え後ベクトル $\mathbf{w}_l^{(a')}$ が生成される。

Combiner の処理では、最大 a 個のベクトル $\mathbf{w}_l^{(a')}$ が排他的論理和を出力する論理演算子に入力される。出力される統合後ベクトルは式 (3.3) を書き換え以下の式で定義される。

$$\mathbf{s}_l = \mathbf{w}_l^{(1)} \oplus \cdots \oplus \mathbf{w}_l^{(a)} \in \mathbb{F}_2^M \quad (3.41)$$

最後に Accumulator による符号化は式 (3.4) によりパリティパケット \mathbf{p}_l が求められる。ここで注意したいことは、Accumulator の処理において、ベクトル \mathbf{s}_{l-1} の最後尾の要素とベクトル \mathbf{s}_l の先頭要素も同様に符号化されることである。ただし、 $l = 1$ のときはこの処理を除く。

以上の符号化を $l \in \{1, \dots, L\}$ に対して行うことにより空間結合 RA 符号の符号語パケットが生成され、以下の式で定義される。

$$\mathbf{C} = \begin{bmatrix} \mathbf{U} & \mathbf{P} \end{bmatrix} \quad (3.42)$$

ただし、 $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$, $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_L]$ である。また符号語パケット \mathbf{C} は先頭に情報パケット \mathbf{U} が付加されていることから、組織符号であると言える。

(2) 検査行列表現

本章で説明してきた空間結合 RA 符号の符号化器の構造から定義される空間結合 RA 符号の検査行列を説明する。

この符号の検査行列 \mathbf{H}_{SC-RA} は以下の式で定義される。

$$\mathbf{H}_{SC-RA} = \begin{bmatrix} \mathbf{\Pi} & \mathbf{A} \end{bmatrix} \quad (3.43)$$

行列 \mathbf{A} は第 (2) 節の式 (3.8) で定義した帯対角行列であり、第 (1) 節において各統合後系列 \mathbf{s}_l は独立ではなく、ベクトル \mathbf{s}_{l-1} の最後尾の要素とベクトル \mathbf{s}_l の先頭要素は符号化されると定義したことにより、RA 符号の行列 \mathbf{A} のサイズはパリティベクトルの数だけ増加し、行列サイズは $LM \times LM$ となる。

一方、一つの例として各プロトグラフ上の Accumulator の処理が独立であると定義すると、その行列は以下のように帯対角行列 \mathbf{A} の対角行列として表される。

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & & & & \\ & \mathbf{A} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \mathbf{A} \\ & & & & & \mathbf{A} \end{bmatrix} \quad (3.44)$$

ただし、行列 \mathbf{A} のサイズは $M \times M$ であり、 L 個の行列 \mathbf{A} が対角に並んだ構造を示す。

次に行列 Π は 2 元 $LM \times KM$ 行列であり，次式で定義される．

$$\Pi = \begin{bmatrix} H_1^{(1)} & & & & & & & \\ H_2^{(1)} & H_2^{(2)} & & & & & & \\ \vdots & \vdots & \ddots & & & & & \\ H_Q^{(1)} & H_Q^{(2)} & \ddots & & & & & \\ & H_{Q+1}^{(1)} & \ddots & & & & & \\ & & \ddots & H_{L-Q-1}^{(a)} & & & & \\ & & & \ddots & H_{L-Q}^{(a-1)} & H_{L-Q}^{(a)} & & \\ & & & & \ddots & \vdots & \vdots & \\ & & & & & H_{L-1}^{(a-1)} & H_{L-1}^{(a)} & \\ & & & & & & H_L^{(a)} & \end{bmatrix} \quad (3.45)$$

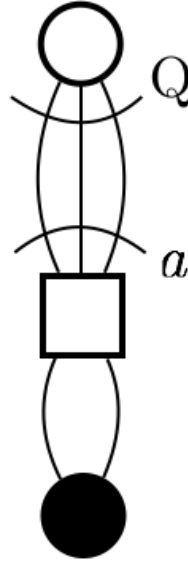
Interleaving 行列 $H_l^{(a')}$ は第 (1) 節において定義されたものである．行列 Π を列方向にみると Q 個の行列 $H_l^{(a')}$ が縦に並び，情報パケット \mathbf{u}_k が Q 回繰り返されていることを示す．また行方向にみると $a'' \in \{1, \dots, a\}$ 個の行列 $H_l^{(a')}$ が横に並び，Combiner において情報パケット $\mathbf{u}_l^{(a'')}$ が統合されていることを示す．また，ある一つの Interleaving 行列 $H_l^{(a')}$ に注目したとき，それは情報パケット $\mathbf{u}_l^{(a')}$ の要素の並びを行列 $H_l^{(a')}$ によって，列置換することに対応する．

3.2.2 プロトグラフ表現

本節では，空間結合 RA 符号のグラフ表現としてプロトグラフを導入する．一般的に，空間結合 RA 符号のタナグラフはノード数が膨大になり，その符号のグラフ構造を理解することが難しい．そこで，空間結合 RA 符号の複数の RA 符号が空間的に近傍するものと結合するという構造を利用して，タナグラフ表現を簡単化したものがプロトグラフ表現である．

(1) RA 符号のプロトグラフ表現

図 3.6 に (Q, a) -regular RA 符号を表現するプロトグラフを示す．このグラフは一つのメッセージノードと一つチェックノード，一つのパリティノードによって表される．そして注目すべきことは，このプロトグラフは第 3.1.2 節の図 3.2 に示される (Q, a) -regular RA 符号のタナグラフと等価であるということである．つまり，プロトグラフ上での一つのメッセージノードはタナグラフ上で M 個のメッセージノードに対応し，一つのチェックノードとパリティノードはタナグラフ上で N 個のチェックノードとパリティノードに対応する．プロトグラフ上のメッセージノードが持つエッジの数はパラメータ Q によって示され Q 本である．つまり，プロトグラフ上での Q 本のエッジはタナグラフ上では，すべてのメッセージノードがそれぞれ Q 本，合計で QM 本のエッジを持っていることになる．同様に，プロトグラフ上でチェックノードが持つメッセージノードと結合するエッジの数は a 本でありタナグラフ上ではそれぞれのチェックノードが a 本，

図 3.6: (Q, a) -regular RA 符号のプロトグラフ表現

合計で aN 本のメッセージノードと結合するエッジを持っていることを示す。また、メッセージノードとチェックノード間に Interleaver が挟まれおり、プロトグラフ上でその記号は省略されている。最後に、プロトグラフ上でのチェックノードとパリティノード間のエッジの結合は、タナーグラフ上で Accumulator の処理に対応するエッジの結合である。

(2) 空間結合 RA 符号プロトグラフ表現

本節では、前節で説明したプロトグラフを用いて空間結合 RA 符号のプロトグラフ表現について述べる。

空間結合は複数のプロトグラフのエッジをお互いに結合することにより行う。図 3.7 は図 3.6 に示す (Q, a) -regular RA 符号のプロトグラフを K 個横に並べたものであり、左からプロトグラフに固有の番号 $k \in \{1, \dots, K\}$ を与える。各プロトグラフのメッセージノードは Q 本のエッジを持っており、そのエッジを右側かつ近傍に存在するプロトグラフ上のチェックノード (自分自身のチェックノードも含む) に 1 本ずつ結合させる。ただし、ある一つのチェックノードに結合できるエッジの数は最大 a 本である。以上の条件のもとエッジを結合すると、図 3.7 上のプロトグラフは図 3.8 のように結合される。このエッジの結合をタナーグラフ上で表現すると、プロトグラフ上の一つのメッセージノードに含まれる M 個のメッセージノードとプロトグラフ上の一つのチェックノードに含まれる M 個のチェックノードがインターリーヴァを挟み結合される。

図 3.8 において注目すべきことは、赤色のプロトグラフが示しているように、図 3.7 と比較してプロトグラフ上のチェックノードとパリティノードがそれぞれ K_{ex} 個増えていることである。これは空間結合を条件通りに行ったときに付加される追加のノードである。よって、空間結合 RA 符号においては追加のノードの数が符号化率 r_{SC-RA} を決定し、詳しくは後ほど説明する。

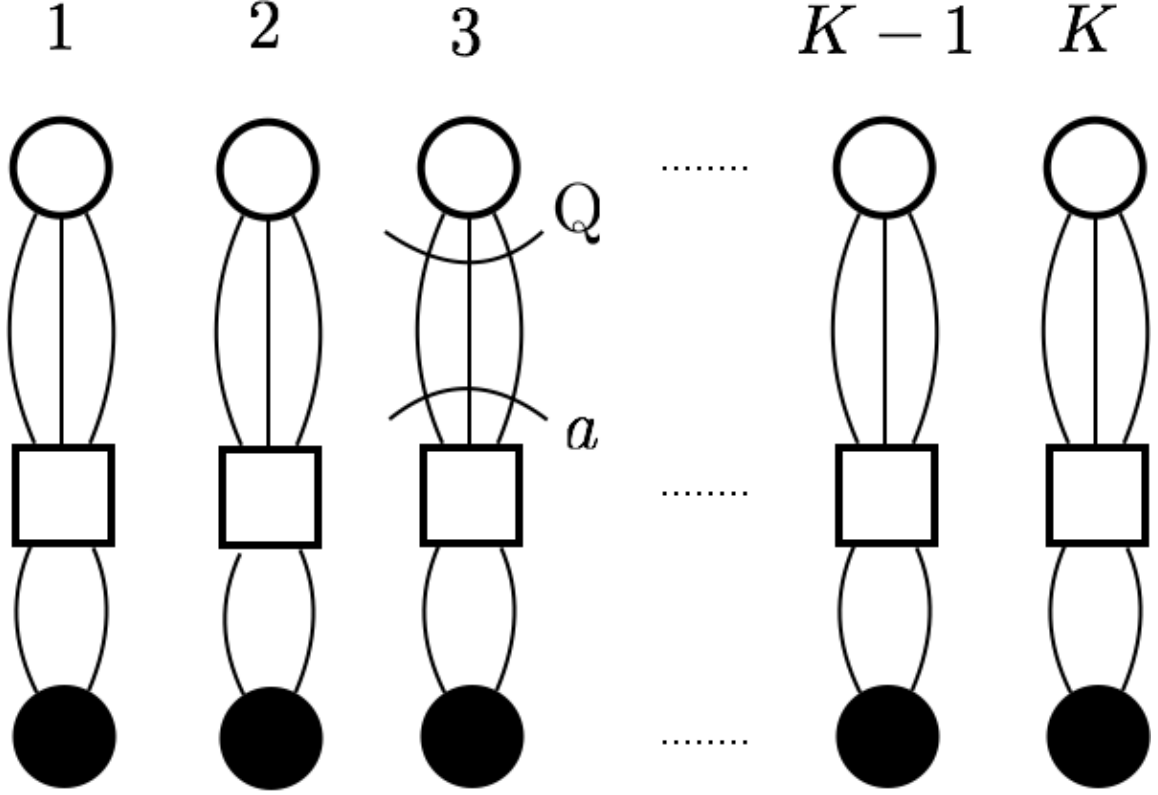

 図 3.7: (Q, a) -regular RA 符号のプロトグラフを K 個コピーし並べたもの

図 3.8 における L はプロトグラフ上でのチェックノードもしくはパリティノードの数を表し、以下の式で定義される。

$$L = \left\lceil \frac{Q}{a} K \right\rceil + (a - 1) \quad (3.46)$$

ただし、 $\lceil x \rceil$ は $x \in \mathbb{R}$ に対して x 以上の最小の整数と定義する。この式は各プロトグラフのメッセージノードが Q 本のエッジを右側かつ近傍に存在するプロトグラフ上のチェックノード (自分自身のチェックノードも含む) に 1 本ずつ結合させ、ある一つのチェックノードに結合できるエッジの数は最大 a 本であるという条件を満たすときに限り、すべての Q と a について成り立つ。

以上から、追加のチェックノードもしくはパリティノードの数は以下の式により与えられる。

$$K_{ex} = L - K = \left\lceil \frac{Q - a}{a} K \right\rceil + a - 1 \quad (3.47)$$

またプロトグラフ上のチェックノードとパリティノードにもメッセージノードと同様に左から固有の番号 $l \in \{1, \dots, L\}$ が与えられるとし、プロトグラフ上のチェックノード $l-1$ のタナーグラフ上での最後尾のノードとパリティノード l のタナーグラフ上での先頭ノードはエッジでつながっているとす。以上が、 (Q, a, K) -空間結合 RA 符号のプロトグラフ表現である。

次に、空間結合 RA 符号のプロトグラフ表現と符号化器に基づく符号化法との関係について述べる。まず、各プロトグラフ上のメッセージノード k は符号化されるメッセージベクトル \mathbf{u}_k に対

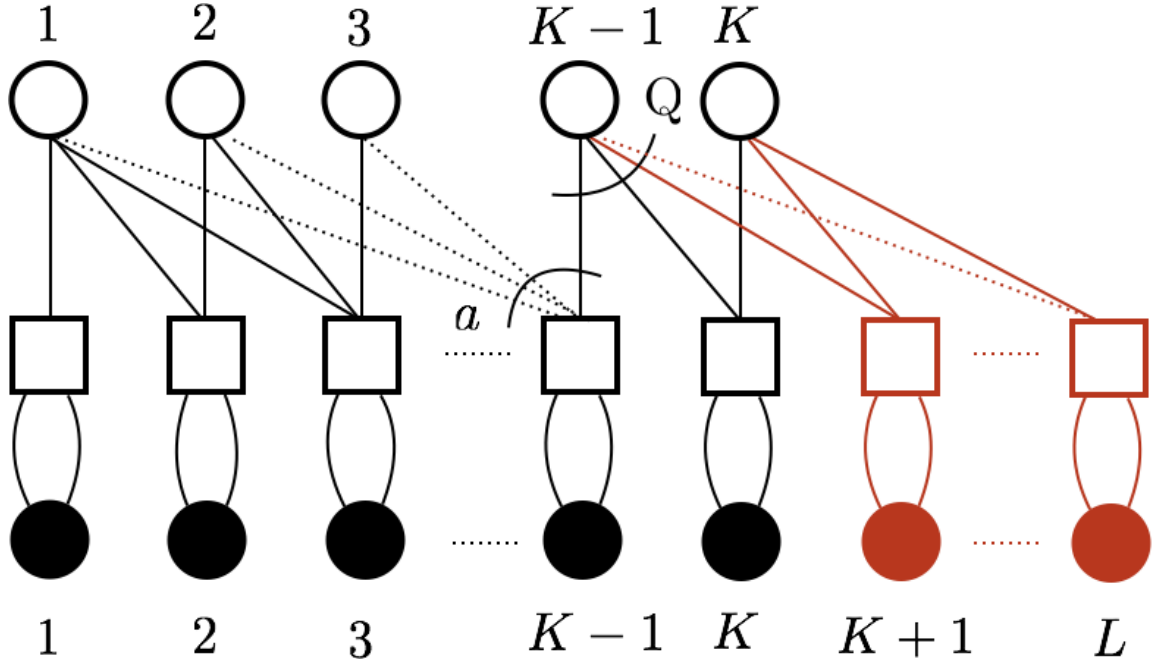


図 3.8: 空間結合 RA 符号のプロトグラフ表現

応する．メッセージノード k から出る Q 本のエッジはメッセージベクトル \mathbf{u}_k が Q 回繰り返されていることを示し，各エッジに対応する Interleaving 行列 $H_l^{(a')}$ が各メッセージベクトル $\mathbf{u}_k^{(a')}$ の要素の並びをランダムに入れ替える．そして，繰り返されたメッセージノードのエッジはチェックノード l と結合される．これは入れ替え後ベクトル $\mathbf{w}_l^{(a')}$ がチェックノード l に入力され，排他的論理和をとることに対応している．このチェックノード l 上で生成されるものが式 (3.3) で表される統合後ベクトル \mathbf{s}_l である．このチェックノード $l \in \{1, \dots, L\}$ 上のベクトルに対して，Accumulator の処理が行われ下部の各パリティノードに対応したパリティベクトル \mathbf{p}_l が生成される．そして，符号語 \mathbf{C} は各メッセージノードに対応したメッセージベクトル \mathbf{u}_k と各パリティノードに対応したパリティベクトル \mathbf{p}_l によって構成される．

(Q, a, K) -空間結合 RA 符号プロトグラフ表現から，プロトグラフ上のメッセージノードの数は K 個であり，一つのプロトグラフに含まれるメッセージベクトルの要素数は M であるので，空間結合 RA 符号のメッセージベクトルの全要素数は KM である．同様にパリティノードの数は L 個であり，一つのプロトグラフに含まれるパリティベクトルの要素数は M であるので，パリティベクトルの全要素数は LM となる．以上から，空間結合 RA 符号の符号化率は次式で求まり，

$$r_{SC-RA} = \frac{KM}{KM + LM} = \frac{K}{K + L} = \frac{K}{\left\lceil \frac{Q+a}{a} K \right\rceil + a - 1} \quad (3.48)$$

符号化率はパラメータ Q, a, K のみによって決定される．この符号化率を持つ符号を (Q, a, K) -空間結合 RA 符号と定義される．

さらに、メッセージノード数 K を無限大にしたとき符号化率は

$$\lim_{K \rightarrow \infty} r_{SC-RA} = \frac{a}{Q + a} \quad (3.49)$$

となる。

3.2.3 復号法

本章では、空間結合 RA 符号の復号を第 (4) 節で示した対数領域 sum-product 復号法と、第 (2) 節の図 3.8 に示した空間結合 RA 符号のプロトグラフ表現を用いて復号を行う。

まず、図 3.8 のプロトグラフをタナグラフ上で考え、メッセージノードには $u(k_m)$ $m \in \{1, \dots, M\}$ 、チェックノードは $f(l_m)$ 、パリティノードは $p(l_m)$ の固有の番号をそれぞれ与える。ただし、 k と l は第 (2) 節で定義したプロトグラフ上での各ノードの固有の番号である。

次に、第 3.1.2 節と同様に、メッセージノード $u(k_m)$ と結合するチェックノードの集合を $D(u(k_m))$ 、チェックノード $f(l_m)$ と結合するメッセージノードとパリティノードの集合を $E(f(l_m))$ と定義する。

さらに、受信信号ベクトル \mathbf{Y} は

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_{K+L} \end{bmatrix} \in \mathbb{R}^{(K+L)M} \quad (3.50)$$

となり、各受信信号ベクトル \mathbf{y}_i は第 2 章の式 (2.3) により定義される。

以上の定義のもと、第 (4) 節の対数領域 sum-product 復号法を実行することにより所望の推定語ベクトル $\hat{\mathbf{C}}$ が導出される。ただし、推定語ベクトル $\hat{\mathbf{C}}$ は以下の式で表される。

$$\hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{U}} & \hat{\mathbf{P}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{u}}_1 & \dots & \hat{\mathbf{u}}_K & \hat{\mathbf{p}}_1 & \dots & \hat{\mathbf{p}}_L \end{bmatrix} \in \mathbb{F}_2^{(K+L)M} \quad (3.51)$$

3.3 多次元空間結合 RA 符号化協調

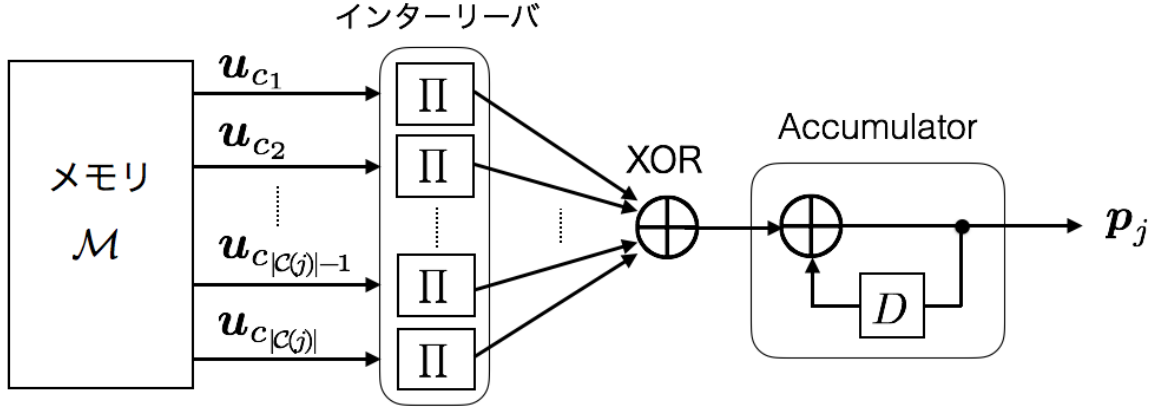
MD-SC-RA-CC は SC-RA-CC の一般形であると考えられるため、本節では MD-SC-RA-CC の構成法を中心に説明する。

3.3.1 符号化法

図 3.9 に、スロット j における端末 i の符号化器を示す。符号化器は、情報パケットを W 個格納可能な 1 つのメモリ \mathcal{M} と、 Q 個のお互いに統計的に独立なインターリーバ Π 、排他的論理和演算子、1 ビット遅延器 D と排他的論理和演算子から成る Accumulator によって構成される。

スロット j において、メモリ \mathcal{M} はスロット j 以前に受け取った W 個の情報パケット $\{\mathbf{u}_{j'} \mid j-W \leq j' \leq j-1, j' \in \mathbb{N}\}$ を格納している。ただし、 \mathbb{N} は 1 以上の自然数の集合を表す。第 2 章で述べたように、パリティパケット \mathbf{p}_j はメモリからスロット集合 $\mathcal{C}(j)$ に対応する情報パケットが出力することにより生成される。MD-SC-RA-CC において、集合 $\mathcal{C}(j)$ は以下の式で定義される。

$$\mathcal{C}(j) = \{\{j - v_1, \dots, j - v_q, \dots, j - v_Q\} \cap \mathbb{N}\} \quad (3.52)$$


 図 3.9: スロット j における端末 i の符号化器 (ただし, $\mathcal{C}(j) = \{c_1, c_2, \dots, c_{|\mathcal{C}(j)|}\}$)

ただし, $q = 1, 2, \dots, Q$ である. 集合 $\mathcal{C}(j)$ は \mathbb{N} との集合積を含んでいるため, そのサイズは $0 \leq |\mathcal{C}(j)| \leq Q$ となり, $|\mathcal{C}(j)| = 0$ のとき, パリティパケットは生成されない. パラメータ v_q は, スロット j のパリティパケットが v_q 以前に受け取った情報パケットから生成されることを意味する. これらのパラメータを結合ベクトル \mathbf{v} として, 以下のように定義する.

$$\mathbf{v} = (v_1, \dots, v_q, \dots, v_Q) \quad (3.53)$$

v_q は集合 $\{1, \dots, W\}$ の値から重複なく与えられ, 各要素は昇順に並べられる. ただし, W は v_Q が取り得る最大値を与え, 符号化器のメモリのサイズに対応する.

集合 $\mathcal{C}(j)$ はスロット番号 j の関数であるため, 結合ベクトル \mathbf{v} の値を与えることによって MD-SC-RA-CC の符号化構造を決定することができる. 今後の説明の簡単のため, スロット j において送信される情報パケットがパリティパケットの生成に用いられるときのスロット集合を $\mathcal{V}(j)$ とし, 以下の式で定義する.

$$\mathcal{V}(j) = \{j + v_1, \dots, j + v_q, \dots, j + v_Q\} \quad (3.54)$$

ただし, 集合のサイズ $|\mathcal{V}(j)|$ は常に Q となる.

次に, メモリ \mathcal{M} から出力されたスロット集合 $\mathcal{C}(j)$ に対応する情報パケットはインターリーバ Π に入力される. 本研究において, インターリーバは S -ランダムインターリーバ [26] を用いる. インターリーバから出力された $|\mathcal{C}(j)|$ 個のビット系列は排他的論理和演算子に入力され, 長さ $M[\text{bits}]$ の 1 つの系列が出力される. その後, Accumulator に入力され, パリティパケット \mathbf{p}_j が出力される. 最後に, 情報パケット \mathbf{v}_j とパリティパケット \mathbf{p}_j が接続され符号語となり, ブロードキャスト送信される.

ここで, SC-RA-CC の符号化器について説明する. 結合ベクトル $\mathbf{v} = (1, \dots, Q)$ である MD-SC-RA-CC は, SC-RA-CC に対応すし, スロット j のパリティパケットは, スロット $j - Q$ 以降に受け取った情報パケットから必ず生成されることを意味する. 以上から, SC-RA-CC は MD-SC-RA-CC の特殊形であると言える.

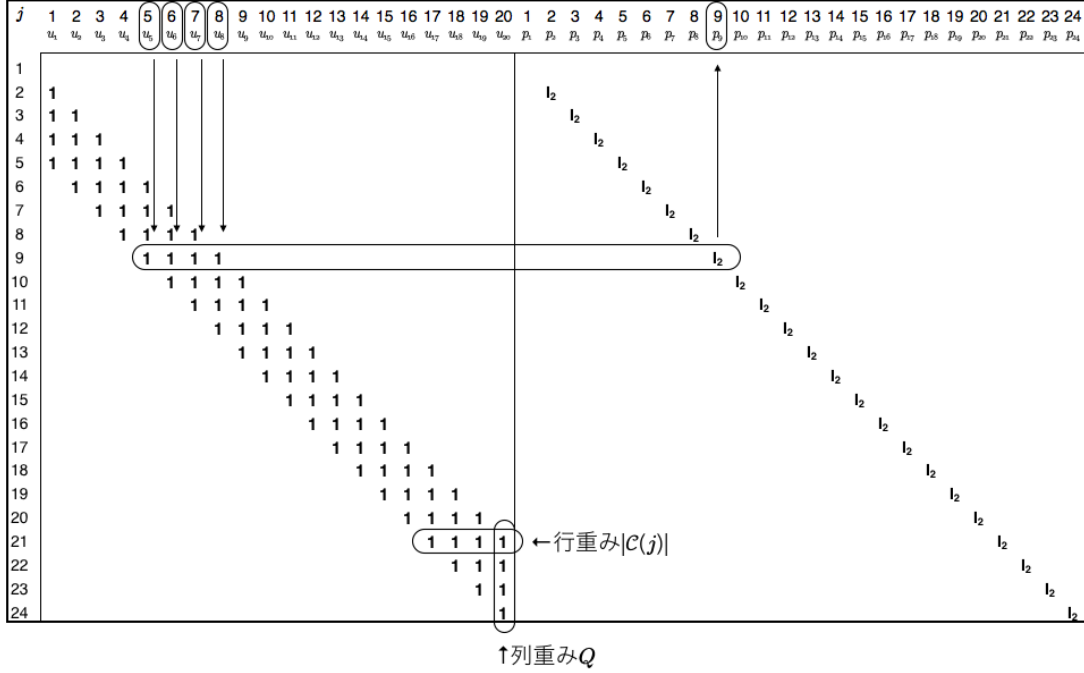


図 3.10: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC のパリティ検査行列

3.3.2 パリティ検査行列

本節において, MD-SC-RA-CC のパリティ検査行列を説明する. 図 3.10 に, 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ のときのパリティ検査行列を示す. 結合ベクトル \mathbf{v} の値から, これは SC-RA-CC に対応する. パリティ検査行列は要素 $\mathbf{1}$ によって構成されるサイズ $(NT + W) \times NT$ の左部分行列と, \mathbf{I}_2 によって構成されるサイズ $(NT + W) \times (NT + W)$ の右部分行列から成る. 各列と行には前述のスロット番号が記載され, 左部分の各列は情報パケットの系列, 右部分の各列はパリティパケットの系列にそれぞれ対応する. 要素 $\mathbf{1}$ は, サイズ $M \times M$, 行重み 1, 列重み 1 の二値行列であり, 各要素 $\mathbf{1}$ はお互いに統計的に独立である. これは符号化器のインターリーバに対応する. 要素 \mathbf{I}_2 はサイズ $M \times M$, 行重み 2, 列重み 2 で幅 2 の 2 値帯行列を表し, これは符号化器の Accumulator の符号化に対応する. パリティ検査行列全体に注目すると, 左部分行列は要素 $\mathbf{1}$ の帯行列として記述され, その行重みは $|C(j)|$, 列重みは Q となる. 右部分行列は, 要素 \mathbf{I}_2 の単位行列として記述される. ただし, スロット 1 においてパリティパケット \mathbf{p}_1 の系列と関係する情報パケットの系列は 1 つも存在しないため, パリティパケット \mathbf{p}_1 の系列は生成されない.

例としてスロット 9 に注目したとき, パリティパケット \mathbf{p}_9 は, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$ と式 (3.52) からスロット集合 $C(9) = \{9 - 1, 9 - 2, 9 - 3, 9 - 4\} = \{5, 6, 7, 8\}$ の情報パケットを用いて生成される. 図 3.10 でも同様に, スロット番号 9 のパリティパケット系列はスロット番号 5, 6, 7, 8 の情報パケット系列と, スロット番号 9 の行を介して, 関係付けられている. 以上から,

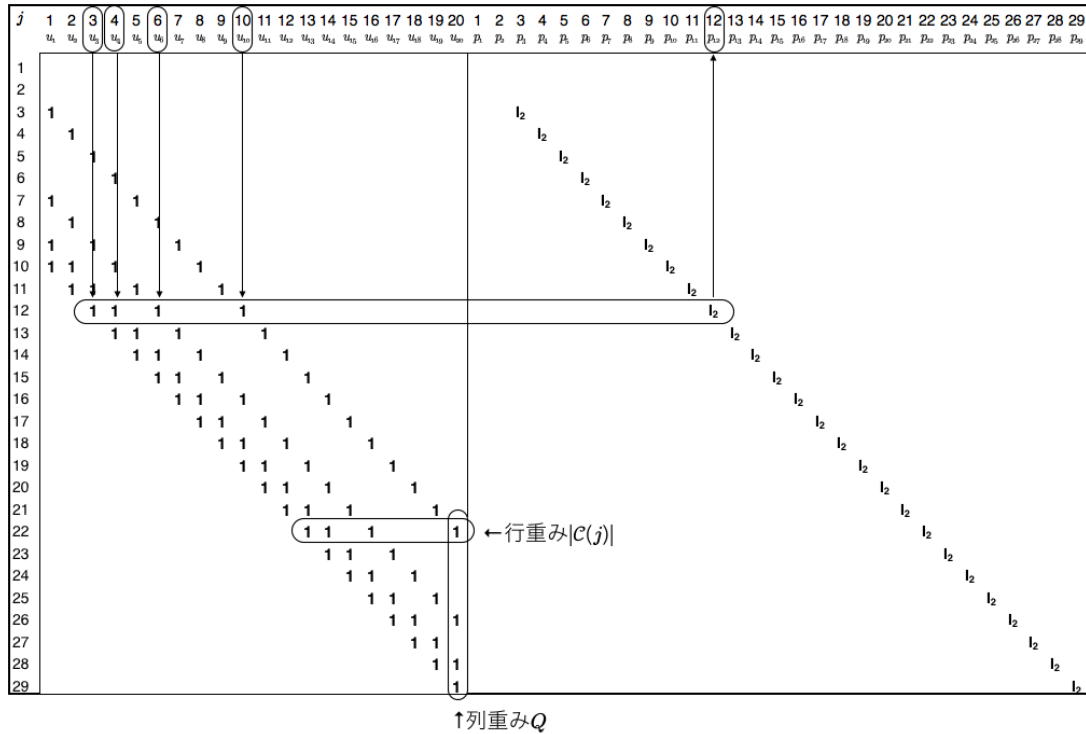


図 3.11: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 9$ である MD-SC-RA-CC のパリティ検査行列

MD-SC-RA-CC のパリティ検査行列の構造も同様に、結合ベクトル \mathbf{v} によって、一意に決定されることがわかる。

図 3.11 に、結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$ 、メモリサイズ $W = 9$ によって定義される MD-SC-RA-CC のパリティ検査行列を示す。結合ベクトル \mathbf{v} の要素の値を SC-RA-CC よりも大きくすることによって、左部分の要素 1 が下側に広がり、5 行分追加されていることがわかる。それに伴い、右部分のスロット $25, \dots, 29$ に注目すると 5 個のパリティパケット系列が追加されていることがわかる。例としてスロット 12 に注目したとき、パリティパケット系列 \mathbf{p}_{12} は、スロット集合 $\mathcal{C}(12) = \{12 - 2, 12 - 6, 12 - 8, 12 - 9\} = \{3, 4, 6, 10\}$ の情報パケット系列によって制約を受けていることがわかる。また、式 (3.52) より、スロット 1 と 2 において、 $|\mathcal{C}(1)| = 0, |\mathcal{C}(2)| = 0$ であるため、パリティパケットは生成されない。

3.3.3 グラフ表現

本節において、MD-SC-RA-CC のグラフ表現を説明する．今回，説明するグラフは，タナーグラフを簡易的に表したプロトグラフであり，これはパリティ検査行列から一意に決定される．

図 3.12 に、全情報パケット数 $NT = 20$ 、結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$ 、ベクトルの要素数 $Q = 4$ のときの 1 次元プロトグラフを表す。図 3.10 のパリティ検査行列と同様に、結合ベクトル \mathbf{v} の値から、これは MD-SC-RA-CC を特殊化した SC-RA-CC に対応する。グラフ上部の数字は、式

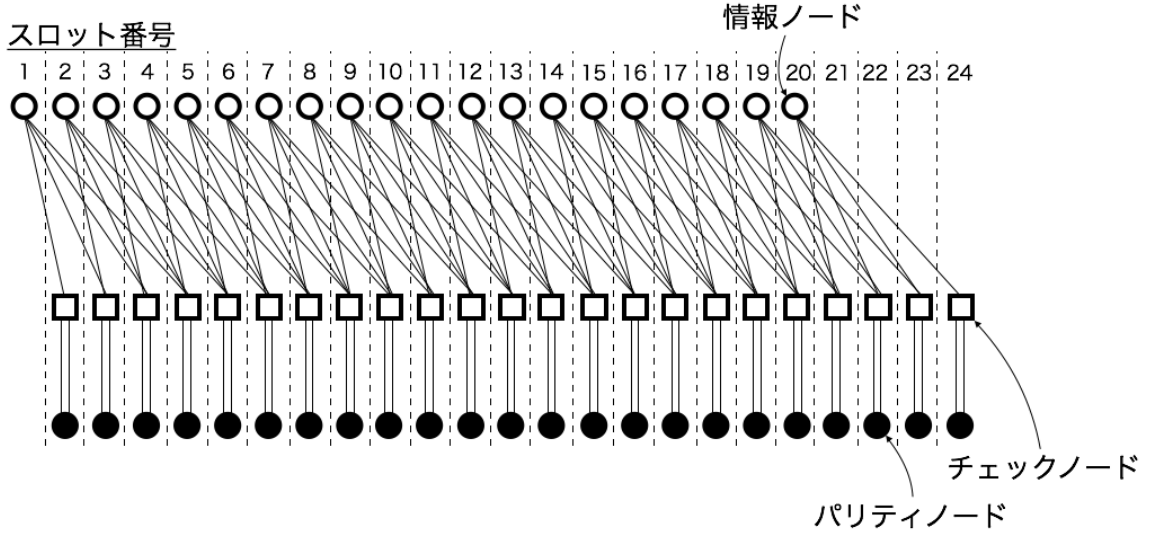


図 3.12: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC の 1 次元プロトグラフ表現

(2.1) のスロット番号を表す。各スロット番号に対応する白丸は情報ノードを表し、検査行列の情報パケット系列、そして情報パケットに対応する。同様に、黒丸はパリティノードを表し、検査行列のパリティパケット系列、そしてパリティパケットに対応する。白四角はチェックノードと呼ばれ、検査行列の行成分を意味し、そして対応するパリティパケットの制約条件を表す。

情報ノードとチェックノード間を結ぶエッジは、パリティ検査行列の要素 $\mathbf{1}$ に対応し、インターリーバを含む。つまり、検査行列上で要素 $\mathbf{1}$ が列 j_c と行 j_r 上に位置するとき、列に対応する情報ノード j_c と行に対応するチェックノード j_r 間でエッジが形成される。ただし、 $j_c = 1, 2, \dots, NT$, $j_r = 1, 2, \dots, NT + W$ である。検査行列の左部分行列において列重みは Q であるため、全ての情報ノードから伸びるエッジの数は $|\mathcal{V}(j)| = Q$ である。一方、スロット j においてチェックノードから情報ノードに伸びるエッジの数は $|\mathcal{C}(j)|$ であり、これも同様に検査行列の左部分の行重みと一致する。

パリティノードとチェックノード間の平行な 2 本のエッジはパリティ検査行列の要素 \mathbf{I}_2 に対応し、Accumulator の符号化を表す。

ここで、スロット 5 のパリティノードに注目する。これはチェックノードを介してスロット 1, 2, 3, 4 の情報ノードと結合している。注目するパリティノードに対応するパリティパケットが、結合する情報ノードに対応する情報パケットから生成されることを意味する。MD-SC-RA-CC のプロトグラフ構造は、各スロット j におけるチェックノードから情報ノードへのエッジの集合 $\mathcal{C}(j)$ によって定義され、 $\mathcal{C}(j)$ は結合ベクトル \mathbf{v} によって決定される。よって、全情報パケット数 NT とベクトルの要素数 Q , メモリサイズ W を固定したとき、MD-SC-RA-CC のプロトグラフ構造は、結合ベクトル \mathbf{v} によって一意に決められる。

図 3.11 に、結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, メモリサイズ $W = 9$ の MD-SC-RA-CC の 1 次元プロトグラフを示す。パリティ検査行列と同様に、結合ベクトル \mathbf{v} の要素が大きくなることによっ

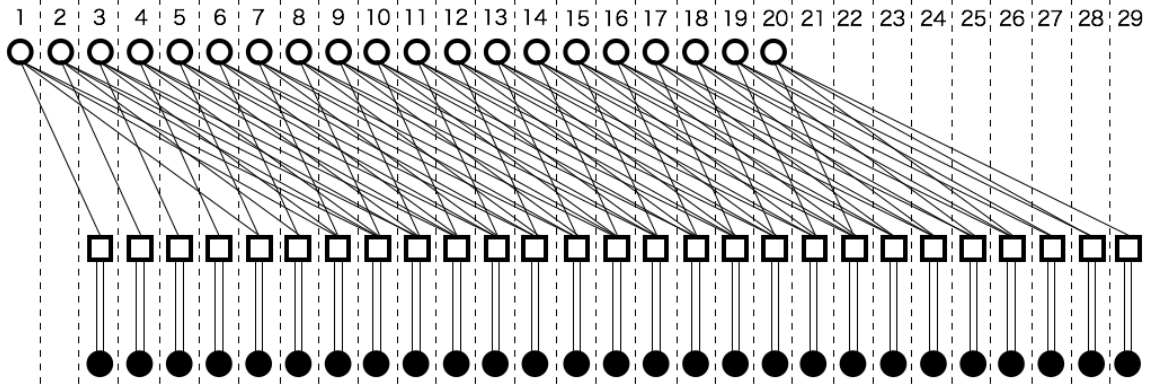


図 3.13: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 9$ である MD-SC-RA-CC の 1 次元プロトグラフ表現

て、プロトグラフの右側に追加のパリティノードとチェックノードが 5 個ずつ付加されていることがわかる。

次に、上記の 2 つのプロトグラフ表現を、2 次元グラフに書き換えたプロトグラフ表現を図 3.14 と図 3.15 に示す。これまでは、スロット番号 j を用いて議論していたが、ここでは端末番号 i とブロック番号 t を導入し、それらを用いた 2 次元グラフ表現で表す。図の縦軸は端末番号 i に対応し、横軸はスロット番号 t に対応する。つまり、図 3.12 と図 3.13 のプロトグラフを $N = 5$ ごとに折り返したものとなっている。図 3.14 と図 3.15 を比較すると、前者において情報ノードは最も近傍に存在する Q 個のチェックノードと結合しているのに対して、後者は最大 2 ブロック先のチェックノードとつながっていることがわかる。後者は MD-SC-RA-CC であり、これが多次元空間結合と言われる由縁である。また注目すべき点としては、図 3.14 と図 3.15 とともに、ある情報ノードから伸びるエッジが全て異なる端末番号のチェックノードと結合していることである。これは、各情報パケットと、それを符号化したパリティパケットが異なる端末から基地局に送信されることを意味しており、このような結合状態を、“フルエッジ結合”と定義する。

3.3.4 符号化率

プロトグラフ表現から、各チェックノードには異なる数のエッジが入力されていることがわかる。図 3.12 を見ると、スロット 2 のチェックノードには 1 本の情報ノードのエッジが入力されているのに対して、スロット 10 のチェックノードには $Q = 4$ 本のエッジが入力されている。このように空間結合符号においては、プロトグラフの両端のチェックノードは小さい度数が与えられ、中心に行くほどより大きい度数が与えられる。これは、プロトグラフの両端で生成されたパリティパケットの方が符号化率が小さく、プロトグラフの真ん中に行くほど符号化率が大きくなることを意味する。

MD-SC-RA-CC の復号には、プロトグラフ上での確率伝搬法 [16] が用いられる。確率伝搬法においては、プロトグラフの符号化率が小さい両端から復号がスタートし、その復号が連鎖となり、中心に向かって進んで行く。

一方，プロトグラフの両端の符号化率が低いということは，空間結合符号全体として符号化率が小さくなることを意味する．MD-SC-RA-CC の符号化率は以下の式で与えられる．

$$r = \frac{NTM}{NTM + (NT + v_Q - v_1)M} = \frac{NT}{2NT + v_Q - v_1} \quad (3.55)$$

NTM は全情報パケットのビット数に対応し， $(NT + v_Q - v_1)M$ は全パリティパケットのビット数に対応する．この式から分かるように，端末数 N もしくは，ブロック数 T が無限大のとき， $(v_Q - v_1)$ の項が無視されて， r は $1/2$ に漸近する．一方，メモリサイズ W を無限大にし， $(v_Q - v_1)$ を無限大にすると，MD-SC-RA-CC の符号化率 r は 0 に漸近すると言える．以上から，MD-SC-RA-CC においては，結合を広げることによるバースト誤りに対するロバスト性と符号化率がトレードオフである．ただし MD-SC-RA-CC において符号のレートロスが最も小さい符号化構成は，結合ベクトル $\mathbf{v} = (1, 2, \dots, Q)$ の SC-RA-CC である．

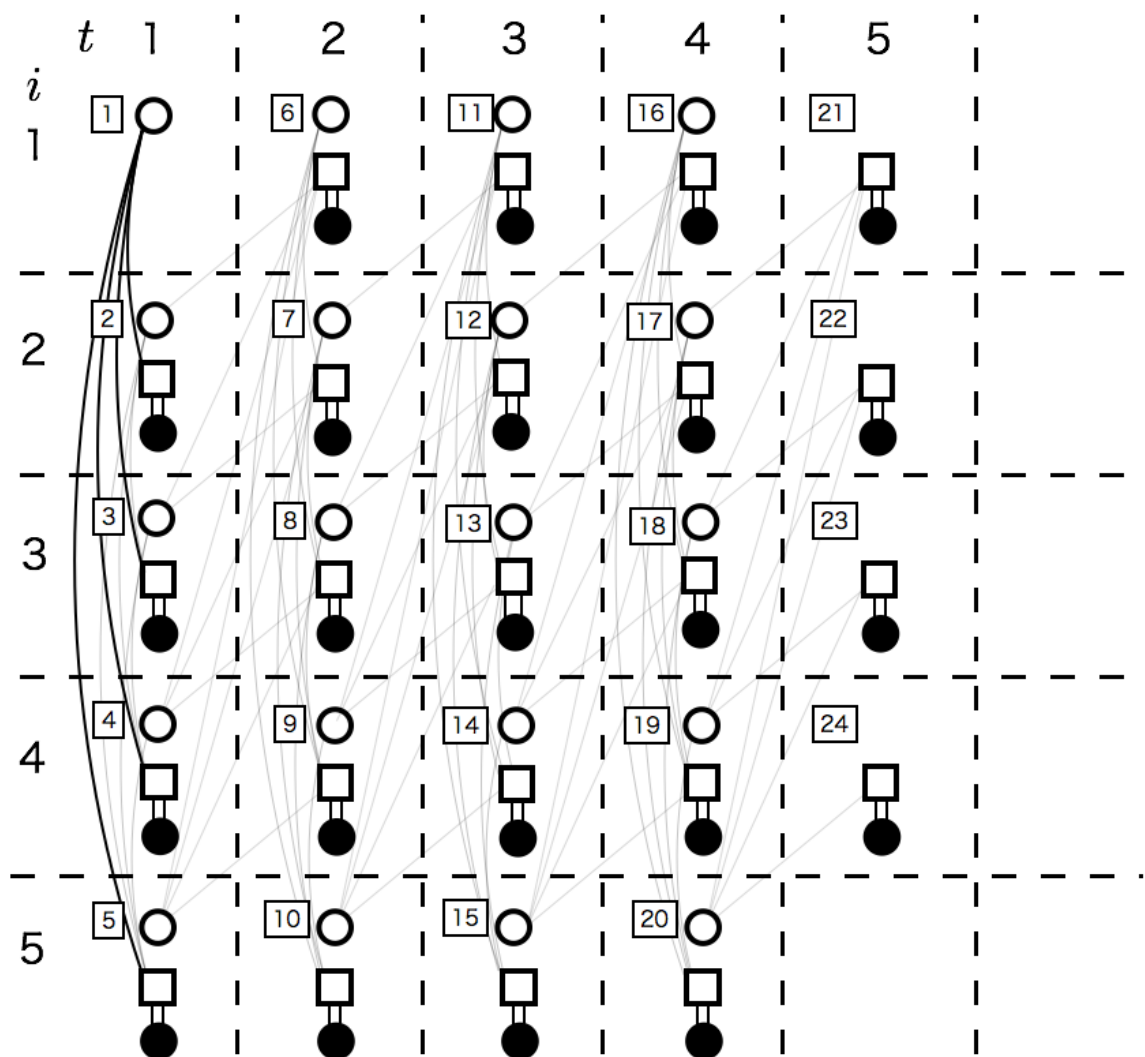


図 3.14: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC の 2 次元プロットグラフ表現

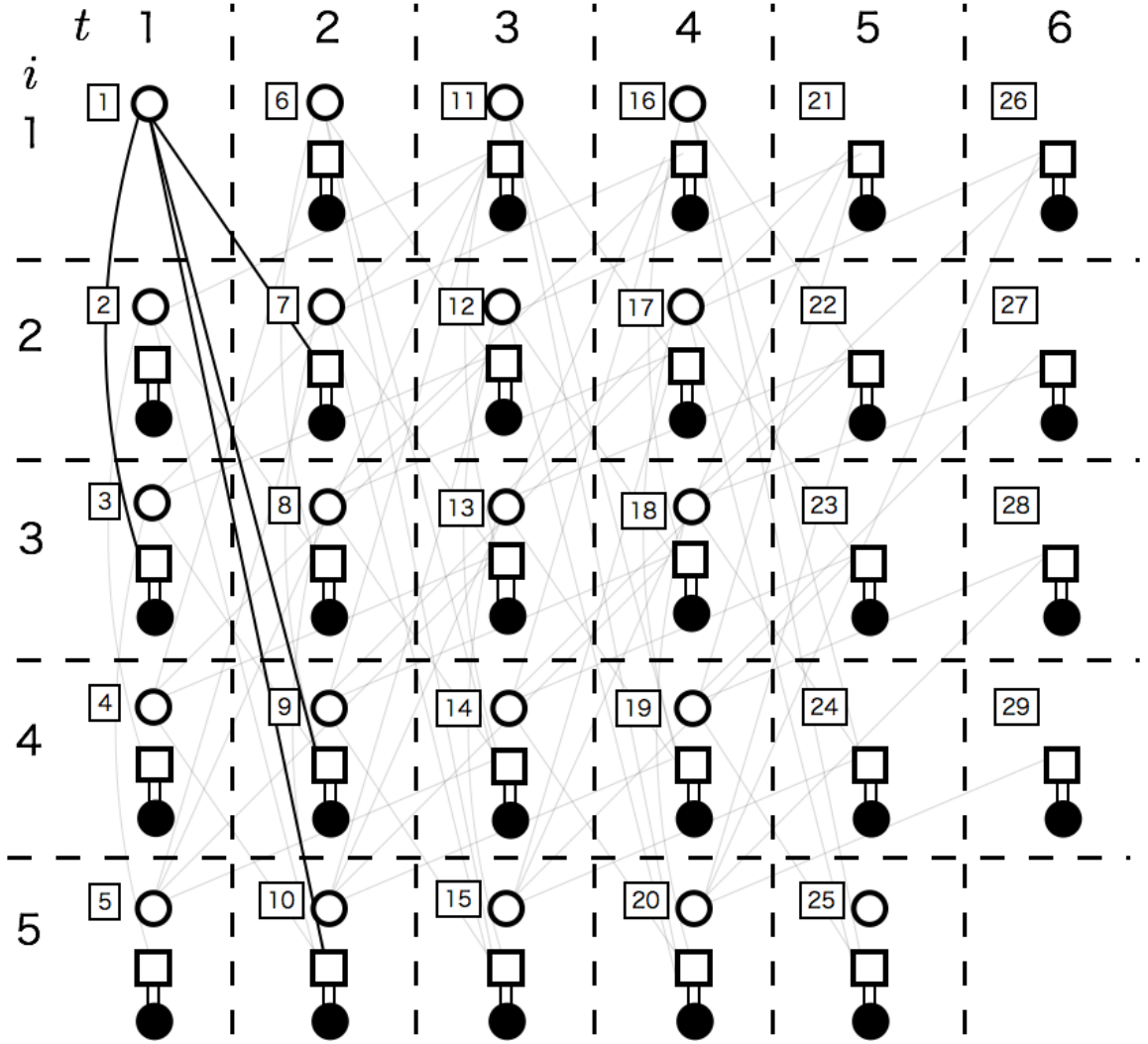


図 3.15: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 9$ である MD-SC-RA-CC の 2 次元プロトグラフ表現

第4章 符号化協調通信の符号化設計

本章では、MD-SC-RA-CCの符号化設計について考える。具体的に述べると、1-ブロックと T -ブロックレイリーフェーディング通信路の両方において優れた復号特性を示す MD-SC-RA-CC の結合ベクトル \mathbf{v} を求めたい。LDPC 符号などの符号において、一般的にその復号特性の解析は密度発展法 (DE: Density Evolution) を用いて行われる [27]。しかしながら、ブロックレイリーフェーディングにおける密度発展法を用いての解析は、演算量の観点で難しい。そこで、本節ではブロックレイリーフェーディングの近似モデルを導入し、その通信路での密度発展法とそのときの下界を導出し、優れた復号特性を示す結合ベクトル \mathbf{v} の導出を行う。

4.1 ブロック消失通信路モデル

ブロックレイリーフェーディングにおいては、一旦通信路が著しく悪化すると、符号語全体の信号の受信信号電力が平均雑音電力以下になってしまい、符号語の誤り率は非常に大きくなってしまう。有限の信号電力対雑音電力比 (SNR: Signal-to-Noise Ratio) においては、このようなイベントの発生確率であるアウテージ確率をゼロにすることはできない。この事実から、ブロックレイリーフェーディング通信路をより簡便に表したブロック消失通信路 (BEC: Block Erasure Channels)[28]を導入する。BEC の定義を以下に示す。

送信されたパケットの全ビットは確率 ϵ_B において全て消失し、確率 $1 - \epsilon_B$ で全て正しく受信される。

システムモデルにおいて、2種類のブロックレイリーフェーディングを定義したように、本節においても 1-BEC と T -BEC の2種類を定義する。1-BEC においては、ブロック t で端末 i が送信したパケットは確率 ϵ_B に従い、統計的独立に消失する。つまり、スロット j ごとに、消失か消失なしかが与えられる。一方、 T -BEC においては、スロット t には関係なく、端末 i のみに対して、パケットは確率 ϵ_B に従い、統計的独立に消失する。つまり、各端末と基地局間の通信路の環境は符号化協調通信の始まりから終了まで一切変わらない。

本章において 1-BEC と T -BEC の両通信路において優れた復号特性を示す MD-SC-RA-CC の結合ベクトル \mathbf{v} を求める。

4.2 密度発展法

密度発展法は復号時の確率伝搬を追いかけることにより符号の復号特性を解析する手法である。本節では、MD-SC-RA-CC の密度発展法を前節のプロトグラフ上で解釈し説明する。

$x(1)_l^{(j)(q)}$ は、スロット j に位置する情報ノードから伸びる q 番目のエッジについて、情報ノードからチェックノード方向に流れる消失確率を表し、 l は密度発展法における確率伝搬の回数を表す。同様に、 $x(2)_l^{(j)(q)}$ は、スロット j に位置する情報ノードから伸びる q 番目のエッジについて、チェックノードから情報ノード方向に流れる消失確率を表す。今後の説明の簡単化のために、プロトグラフ全体でのエッジ番号 $k = (j-1)Q + q$ を定義する。以下では、スロット j の q 番目の情報ノードのエッジの消失確率を $x(1)_l^{(k)} \triangleq x(1)_l^{(j)(q)}$ 、 $x(2)_l^{(k)} \triangleq x(2)_l^{(j)(q)}$ と表す。 $y(1)_l^{(j)}$ は、スロット j に位置するパリティノードから伸びるエッジについて、パリティノードからチェックノード方向に流れる消失確率を表し、 $y(2)_l^{(j)}$ はチェックノードからパリティノード方向に流れる消失確率を表す。ただし、 x と y について右肩にかかるインデックスが異なることに注意したい。密度発展法では、確率伝搬法を行うことにより、これらの消失確率の変化を追いかける。

スロット j の情報ノードに入力されるエッジの番号集合 $\mathcal{V}(j)$ を以下のように定義する。

$$\mathcal{V}(j) = \{(j-1)Q + 1, \dots, (j-1)Q + Q\}. \quad (4.1)$$

次に、スロット j のチェックノードに入力されるエッジの番号集合 $\mathcal{C}'(j)$ を以下のように定義する。

$$\mathcal{C}'(j) = \{(j-v_1)Q + Q, \dots, (j-v_Q)Q + 1\} \cap \mathbb{N}. \quad (4.2)$$

さらに、スロット j において送信されたパケットが伝搬する通信路のビット消失確率を $\epsilon^{(j)}$ とする。ただし、BEC においては、スロット j における通信路のビット消失確率は $\{0, 1\}$ の 2 通りであり、1 となる確率がブロック消失確率 ϵ_B によって与えられる。

$l+1$ 回目の確率伝搬におけるチェックノードから情報ノードへのエッジが持つ消失確率 $x(2)_{l+1}^{(k)}$ は次式で与えられる。

$$x(2)_{l+1}^{(k)} = \left\{ 1 - \left(1 - y(1)_l^{(j)}\right)^2 \prod_{n \in \mathcal{C}'(j) \setminus k} \left(1 - x(1)_l^{(n)}\right) \right\} \quad (4.3)$$

これは、スロット j のチェックノードについて、エッジ k を除く全てのエッジの消失でない確率を乗算したものの補事象を表す。パリティノードからチェックノードへ伸びるエッジは 2 本あるため、 $(1 - y(1)_l^{(j)})$ は 2 乗される。同様に、チェックノードからパリティノードへのエッジが持つ消失確率 $y(2)_{l+1}^{(j)}$ は次式で与えられる。

$$y(2)_{l+1}^{(j)} = \left\{ 1 - \left(1 - y(1)_l^{(j)}\right) \prod_{n \in \mathcal{C}'(j)} \left(1 - x(1)_l^{(n)}\right) \right\} \quad (4.4)$$

最後に、 $l+1$ 回目の確率伝搬における情報ノードからチェックノードへのエッジが持つ消失確率 $x(1)_{l+1}^{(k)}$ は次式で与えられる。

$$x(1)_{l+1}^{(k)} = \epsilon^{(j)} \cdot \prod_{m \in \mathcal{V}(j) \setminus k} x(2)_{l+1}^{(m)} \quad (4.5)$$

これは、スロット j の情報ノードについて、エッジ k を除く全てのエッジの消失確率を乗算したものに、対応する情報パケットの通信路での消失確率を掛け合わせたものである。同様に、パリティ

ティノードからチェックノードへのエッジが持つ消失確率 $y(1)_{l+1}^{(j)}$ は次式で与えられる．

$$y(1)_{l+1}^{(j)} = \epsilon^{(j)} \cdot y(2)_{l+1}^{(j)} \quad (4.6)$$

パリティノードには 2 本のエッジのみが入力されるため， $x(1)_{l+1}^{(k)}$ とは異なる式になる．しかし，式の本質としてはどちらも同じことを意味している．

本研究では，上記の式で表される密度発展法を用いて，結合ベクトル \mathbf{v} ごとの復号特性を評価する．ただし BEC においては，ビット消失通信路のように消失確率が完全に 0 になる瞬間である閾値を求めることはできず，通信路状態においては必ず復号が停止する場合が存在する．そのため，各通信路状態に対して，MD-SC-RA-CC による復号が失敗する確率を表す復号失敗確率 (DFP: Decoding Failure Probability) P_d を導入し，これを用いて，結合ベクトル \mathbf{v} ごとの評価を行う．

4.3 復号失敗確率の下界

本節では，結合ベクトル \mathbf{v} の復号特性の評価基準として，1-BEC と T -BEC における MD-SC-RA-CC の復号失敗確率の下界を導出する．

4.3.1 1-BEC の場合

1-BEC において前節の密度発展法の確率伝搬が必ず停止するときの通信路状態の条件は以下のようになる．スロット j において情報パケットが消失であり，かつその情報パケットを用いて生成されたすべてのパリティパケットが消失であるときである．各スロットにおけるパケットの消失の有無は統計的に独立であるため，上記のイベントの発生確率は， $\epsilon_B^{(Q+1)}$ で与えられる．さらに，このイベントの発生はすべての情報パケットにおいて統計的独立に考えられる．よって，1-BEC 上における復号失敗確率の下界は次式で表される．

$$P_d \geq 1 - (1 - \epsilon_B^{Q+1})^{NT} \quad (4.7)$$

4.3.2 T -BEC の場合

ビット消失通信路における通信路容量は通信路での消失確率を ϵ と置くと， $1 - \epsilon$ によって表される． T -BEC における通信路容量も同様に求められる．例えば，端末数 $N = 5$ と仮定すると， T -BEC 上で統計的に独立な通信路は 5 本存在し，各通信路に対して消失か消失でないかはブロック消失確率 ϵ_B によって与えられる．5 本の通信路中 2 本の通信路が消失したとき，全体での瞬時的な消失確率は $2/5 = 0.4$ となり， T -BEC における瞬時的な通信路容量は $1 - 0.4 = 0.6$ となる．式 (3.55) から MD-SC-RA-CC の実用的な符号化率は 0.5 より少し小さい値であるため，この通信路状態では復号が可能と言える．一方，5 本の通信路中 3 本の通信路が消失したとき，全体での瞬時的な消失確率は $3/5 = 0.6$ となり， T -BEC における瞬時的な通信路容量は $1 - 0.6 = 0.4$ となる．このとき，通信路容量が MD-SC-RA-CC の符号化率よりも小さくなるため，消失を訂正する

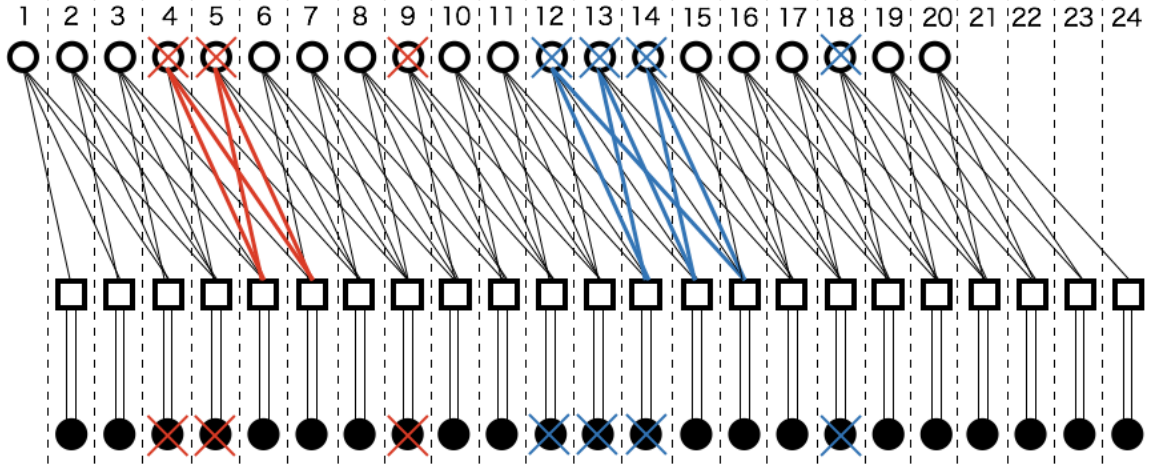


図 4.1: 全情報パケット数 $NT = 20$, 結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルの要素数 $Q = 4$, メモリサイズ $W = 4$ である SC-RA-CC の 1 次元プロトグラフ表現. スロット 4, 5 の情報ノードと、スロット 5, 6 のチェックノード間において長さ 4 のループ, スロット 12, 13, 14 の情報ノードとスロット 14, 15, 16 のチェックノード間において長さ 6 のループが存在

ことができず、復号は失敗する. よって、 T -BEC における復号失敗確率の下界は、通信路容量が MD-SC-RA-CC 符号化率よりも下回らない確率であり、次式によって与えられる.

$$P_d \geq \sum_{l=\lceil(1-r)N\rceil}^N {}_N C_l \epsilon^l (1-\epsilon)^{N-l} \quad (4.8)$$

$\lceil(1-r)N\rceil$ は通信路容量が符号化率を下回るときに、消失している通信路の最小本数を表す. そして上記の式は、通信路が rN 本以上消失する確率を 2 項分布として表している.

ここで、瞬時的な通信路容量が符号化率を下回らないとき、消失している通信路の最大本数は $\lfloor(1-r)N\rfloor$ で表される. このときの瞬時的な通信路容量を”下界における最大の瞬時通信路容量”と定義する.

4.4 符号化構造の設計指針

本節では、復号失敗確率に漸近するような結合ベクトル \mathbf{v} を導出するための、符号の設計指針について考察する.

4.4.1 1-BEC の場合

図 4.1 に結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$, ベクトルサイズ $Q = 4$ である SC-RA-CC のプロトグラフ表現を示す. このプロトグラフは、4 本の赤色のエッジで示される長さ 4 のループを持つ. さらに長さ 4 のループはプロトグラフの至るところで複数見ることができる. この長さ 4 のループが

MD-SC-RA-CC の復号失敗確率を劣化させる理由を説明する．一例として，スロット 4 と 5 における通信路が消失したと仮定する．4.3.1 で説明した復号が必ず停止するイベントの発生条件は，スロット 9 の通信路が消失するときである．このとき，イベントの発生確率は ϵ^3 であり，下界を導出したときの $\epsilon^{Q+1} = \epsilon^5$ よりも明らかに大きいと言える．

次に 6 本の青色のエッジで示される長さ 6 のループに注目する．スロット 12, 13, 14 における通信路が消失したと仮定すると，復号が必ず停止するイベントの発生条件は，スロット 18 の通信路が消失するときである．このイベントの発生確率は ϵ^4 であり，長さ 4 のループのときの発生確率よりは小さいが，下界の導出式における発生確率よりは大きいことがわかる．したがって，1-BEC における MD-SC-RA-CC の符号化設計指針として，長さ 4 のループをできるだけ少なくし，その次に長さ 6 のループを少なくする必要があると言える．今後は長さ 4 と 6 のループを小ループと呼称し，議論を進める．

4.4.2 T -BEC の場合

T -BEC は，符号化協調通信中，通信路の変動が一切ないため，1-BEC よりも厳しい通信路である．ダイバーシティ効果の観点から，情報パケットとそれを符号化したパリティパケットはできるだけ多くの異なる通信路を伝搬する必要がある．図 3.15 に示した MD-SC-RA-CC の 2 次元プロトグラフでは，端末番号 1 に対応するスロット 1 の情報ノードのエッジが，端末番号 2, 3, 4, 5 に対応するチェックノードと結合していることがわかる．つまり，これはスロット 1 で送信された情報パケットとそれを符号化したパリティパケットがお互いに統計的に独立な通信路を伝搬していることを意味する．第 3 章においては，このような状態をフルエッジ結合と定義し，これを数学的に表すと次式で表される．

$$(v_m^{(k)} \bmod N) \neq (v_n^{(k)} \bmod N), (v_m^{(k)} \bmod N) > 0, \forall m, n \in [1, q], m \neq n, N > q \quad (4.9)$$

4.5 BEC における数値例

上記の 2 つの設計指針の意義を明らかにするために，BEC 上での MD-SC-RA-CC の数値結果を示す．ここでは MD-SC-RA-CC のパラメータ条件として，端末数 $N = 5$ ，ブロック数 $T = 20$ ，結合ベクトルサイズ $Q = 4$ であり，所望の符号化率の最小値を $r_{\min} = 0.48$ としたとき， $r > 0.48$ を満たす場合を考える．

この条件のもと，1 つ目の設計指針である，できる限り小ループが少ないプロトグラフを構成する結合ベクトルを求める．文献 [29] の小ループ削減アルゴリズムを用いて，最も小ループが少ない結合ベクトルを導出し，それは $\mathbf{v} = (1, 3, 4, 9)$ となる．次に，2 つ目の設計指針であるフルエッジ条件を満たし，かつその中で最も小ループが少ないプロトグラフを構成する結合ベクトルを求めた．その結果，それは $\mathbf{v} = (2, 6, 8, 9)$ であると明らかになった． $\mathbf{v} = (1, 3, 4, 9)$ は 1-BEC において最も優れた復号特性を示し， $\mathbf{v} = (2, 6, 8, 9)$ は T -BEC において最も優れた復号特性を示し，かつ 1-BEC においても優れた復号特性を示すことが予想できる．

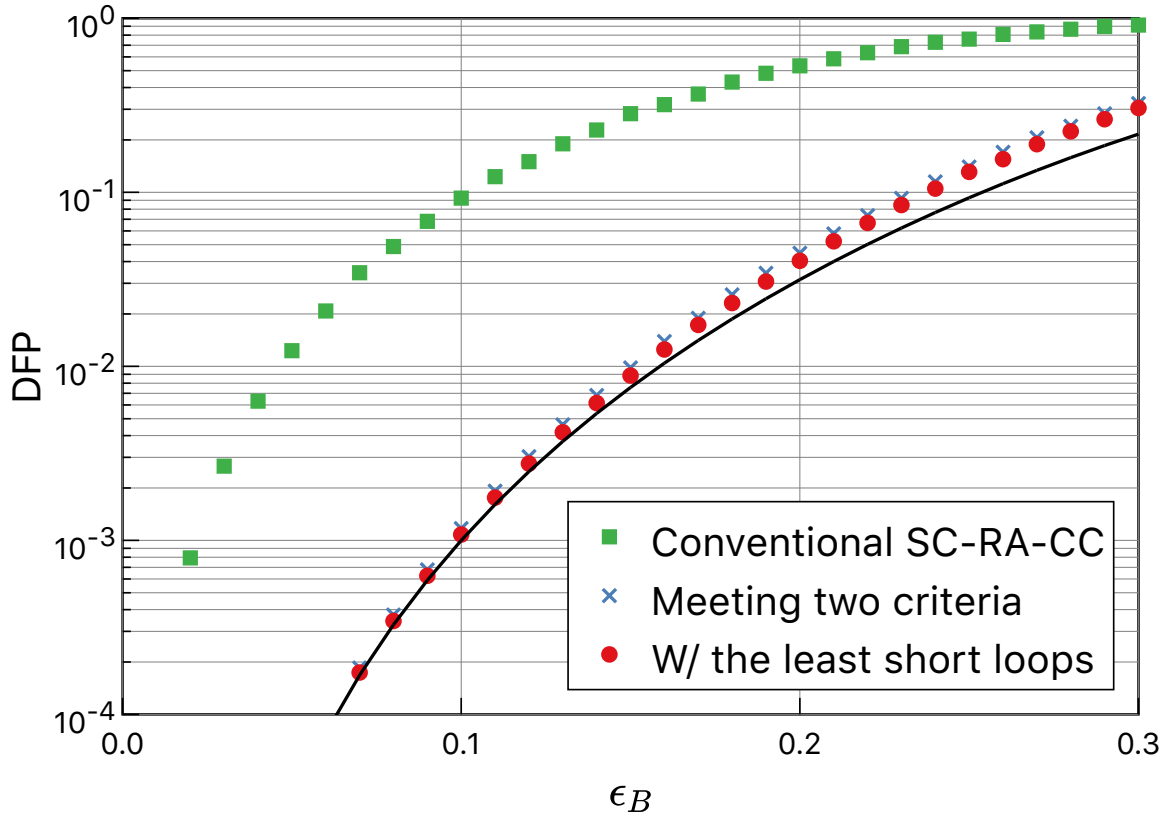


図 4.2: 1-BEC における結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, $(1, 2, 3, 4)$, $(1, 3, 4, 9)$ の MD-SC-RA-CC の DFP 特性 ($N = 5$, $T = 20$, $Q = 4$)

4.5.1 1-BEC の場合

図 4.2 に結合ベクトル $(1, 2, 3, 4)$, $(1, 3, 4, 9)$, $(2, 6, 8, 9)$ によって与えられる MD-SC-RA-CC の 1-BEC 上における DFP 特性を示す。横軸はブロック消失確率 ϵ_B を表し、黒の実線は 1-BEC における DFP 特性の下界を示す。図 4.2 から分かるように、小ループが最も少ないプロトグラフを構成する $\mathbf{v} = (1, 3, 4, 9)$ が DFP の下界に漸近し、最も優れた特性を示すことがわかる。この結果は、1-BEC における設計指針 1 の重要性を明確に示していると言える。 $\mathbf{v} = (2, 6, 8, 9)$ の DFP 特性は、例えば $\epsilon_B = 0.1$ のとき $\mathbf{v} = (1, 3, 4, 9)$ の DFP 特性よりも約 0.001 劣っているが、小ループの数を比較的少なくしているため、DFP の下界に漸近していることがわかる。一方、SC-RA-CC を表す $\mathbf{v} = (1, 2, 3, 4)$ は DFP の下界から常に 0.1 以上離れており、明らかに DFP 特性が悪いことが分かる。これは、SC-RA-CC のプロトグラフ上に、多数の小ループが存在していることが原因であると考えられる。BEC においてはできる限りプロトグラフ上の小ループを除去することが重要であると言える。

表 4.1: T -BEC における結合ベクトル $\mathbf{v} = (2, 6, 8, 9), (1, 2, 3, 4), (1, 3, 4, 9)$ MD-SC-RA-CC の DFP 特性 ($N = 5, T = 20, Q = 4$)

\mathbf{v}	(1, 2, 3, 4)	(1, 3, 4, 9)	(2, 6, 8, 9)	下界
Coding rate	0.493	0.480	0.483	
DFP	4.50×10^{-2}	8.56×10^{-3}	8.56×10^{-3}	8.56×10^{-3}
E_0	0	0	0	0
E_1	0	0	0	0
E_2	5	0	0	0
E_3	10	10	10	10
E_4	5	5	5	5
E_5	1	1	1	1

4.5.2 T -BEC の場合

T -BEC における評価をより合理的に行うため、本節ではまず新しい評価方法について定義する。まず端末数は N であり、端末ごとの各通信路は”消失”と”消失なし”の 2 通りの状態を取るの、 T -BEC において取り得る全ての通信路状態数は 2^N によって与えられる。各通信路状態は、 N 個の通信路のうち l 個の通信路が消失しており、 $N - l$ 個の通信路が消失でない場合が考えられる。ただし、 $l = 0, 1, \dots, N$ 。 l 個の通信路が消失であり、 $N - l$ 個の通信路が消失でない場合において通信路状態の取り得る全ての数は ${}_N C_l$ である。この数に対して、 l 個の通信路が消失であり、かつ MD-SC-RA-CC の復号が失敗するときの通信路状態数を E_l と定義する。したがって、復号が停止する場合の全ての通信路状態数 E_{all} は次式で与えられる。

$$E_{\text{all}} = \sum_{l=0}^N E_l \quad (4.10)$$

ブロック消失確率は ϵ_B であるので、 N 個の通信路のうち l 個の通信路が消失したときの DFP は $\epsilon_B^l (1 - \epsilon_B)^{N-l}$ で表される。したがって、 T -BEC における全通信路状態の DFP は次式で表される。

$$P_D(\mathbf{v}, \epsilon_B) = \sum_{l=0}^N E_l \epsilon_B^l (1 - \epsilon_B)^{N-l} \quad (4.11)$$

ここで、 E_l は結合ベクトル \mathbf{v} によって一意に決定され、これはブロック消失確率は ϵ_B に依存しない。結合ベクトル \mathbf{v} による DFP 特性の差異は、 E_l によってのみ決定され、 ϵ_B に依存しない。以上から、優れた結合ベクトル \mathbf{v} の探索を行うとき、任意のブロック消失確率 $\epsilon_B \in [0, 1]$ において、DFP 特性の評価が可能であると言える。

本節では、 T -BEC において密度発展法を行い、 E_l を求め、結合ベクトル \mathbf{v} の評価を行う。表 4.1 に T -BEC 上において結合ベクトル $(1, 2, 3, 4), (1, 3, 4, 9), (2, 6, 8, 9)$ によって与えられる MD-SC-RA-CC の DFP 特性を示す。ただし、ブロック消失確率は $\epsilon_B = 0.10$ であり、DFP 特性の下

表 4.2: T -BEC における結合ベクトル $\mathbf{v} = (1, 3, 6, 10, 11), (1, 4, 8, 9, 11)$ MD-SC-RA-CC の DFP 特性 ($N = 6, T = 15, Q = 5$)

\mathbf{v}	(1, 3, 6, 10, 11)	(1, 4, 8, 9, 11)	下界
Coding rate	0.480	0.480	
DFP	5.64×10^{-3}	1.27×10^{-3}	1.27×10^{-3}
E_0	0	0	0
E_1	0	0	0
E_2	0	0	0
E_3	6	0	0
E_4	15	15	15
E_5	6	6	6
E_6	1	1	1

界も同時に示す．表 4.1 から， $\mathbf{v} = (1, 2, 3, 4)$ はフルエッジ結合しているにも関わらず $E_2 = 5$ となっており，DFP 特性が下界から乖離していることがわかる．一方 $\mathbf{v} = (1, 3, 4, 9)$ と $(2, 6, 8, 9)$ の DFP 特性は，下界の DFP 特性と完全に一致していることがわかる．この理由は，4.3.2 節で示した下界における最大の瞬時通信路容量と，MD-SC-RA-CC の符号化率との差が大きいことにある．一例として，端末数 $N = 5$ ，符号化率 $r = 0.480$ を仮定すると，下界における最大の瞬時通信路容量は 0.60 となり，これは 5 個の通信路中 2 個の通信路が消失した場合に対応する．よって，符号化率との差は $0.60 - 0.48 = 0.12$ となり，これは 2 つ目の設計指針なしに復号を成功させるためには，十分な差であると考えられる．

そこで，下界における最大の瞬時通信路容量と，MD-SC-RA-CC の符号化率との差が小さい場合を仮定し，MD-SC-RA-CC の評価を行う．ここでは MD-SC-RA-CC のパラメータ条件として，端末数 $N = 6$ ，ブロック数 $T = 15$ ，結合ベクトルサイズ $Q = 5$ であり，所望の符号化率の最小値 $r_{\min} = 0.48$ を満たす場合を考える．このとき，下界における最大の瞬時通信路容量は 0.50 となり，これは 6 個の通信路中 3 個の通信路が消失した場合に対応する．この条件のもと，1 つ目の設計指針である，できる限り小ループが少ないプロトグラフを構成する結合ベクトルは， $\mathbf{v} = (1, 3, 6, 10, 11)$ である．次に，2 つ目の設計指針であるフルエッジ条件を満たし，かつその中で最も小ループが少ないプロトグラフを構成する結合ベクトルは， $\mathbf{v} = (1, 4, 8, 9, 11)$ である．表 4.2 に T -BEC 上において結合ベクトル $(1, 3, 6, 10, 11), (1, 4, 8, 9, 11)$ によって与えられる MD-SC-RA-CC の DFP 特性を示す．ただし，ブロック消失確率は $\epsilon_B = 0.10$ であり，DFP 特性の下界も同時に示す．

表 4.2 から，2 つ目の設計指針を満足している $\mathbf{v} = (1, 4, 8, 9, 11)$ の方が， $E_3 = 0$ となり下界に一致し，DFP 特性において優れていることがわかる．この結果から，2 つ目の設計指針であるフルエッジ結合が T -BEC において DFP 特性を向上させることが明らかになった．

表 4.3: パンクチュアリングパターンごとのビット消失通信路における BP 閾値

	パンクチャド MD-SC-RA-CC			SC-RA-CC
	(1)	(2)	(3)	
符号化率	0.4926			
BP 閾値	0.4768	0.4748	0.4694	0.4867

4.6 パンクチャド MD-SC-RA-CC

式 (3.55) より, MD-SC-RA-CC は SC-RA-CC よりも符号化率が小さくなる. MD-SC-RA-CC と SC-RA-CC の復号特性を比較するとき, その符号化率を等しくして評価を行う必要がある. そのため, 本節においては MD-SC-RA-CC のパリティパケットをパンクチュアリングしたパンクチャド MD-SC-RA-CC について議論する.

4.6.1 パンクチュアリングパターン

MD-SC-RA-CC のパリティパケットをパンクチュアリングするときにおいて, プロトグラフのどここの部分をパンクチュアするかのは非常に重要なテーマである. これを明らかにするために, スロットに対してパンクチュアリングを行う以下の 3 パターンを考える.

1. $\{v_1, \dots, N + v_Q\}$; プロトグラフ全体
2. $\{W - v_1, \dots, N - (W + v_Q)\}$; 両端を除く中側のプロトグラフ
3. $\{v_1, \dots, W - v_1, N - (W + v_Q), \dots, N + v_Q\}$; 中側を除く真ん中のプロトグラフ

ただし, パンクチュアリングは上記の範囲のビットに対して均等に行われるとする.

消失確率 ϵ であるビット消失通信路を仮定し, 3 つのパターンについて 4.2 節の密度発展法を用いて解析することによって, 優れた復号特性を示すパンクチュアリングパターンを求める. 端末数 $N = 5$, ブロック数 $T = 20$, 結合ベクトルサイズ $Q = 4$, 結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$ を仮定する. 評価指標は 以下の式を満たす BP 閾値 ϵ^{th} を用いる.

$$\epsilon^{\text{th}} \triangleq \sup \left\{ \epsilon > 0 \mid \lim_{l \rightarrow \infty} x(2)_l^{(\vee k)} = 0 \right\} \quad (4.12)$$

表 4.3 に密度発展法によって求めた BP 閾値を示す. 表から SC-RA-CC の BP 閾値が最も高いことがわかる. ビット消失通信路のようにブロックごとに消失が発生しない通信路においては, SC-RA-CC が符号構造として最も安定していることが言える. 全てのパンクチュアパターンの中では, (1) が最も優れた BP 閾値を示すことがわかる. ここから, パンクチュアリングパターンは全てのパリティパケットを均等にパンクチュアする必要があると分かる. これは, ビット消失通信路において通信路容量に漸近する LDPC 符号の特徴から明らかな結果である [27].

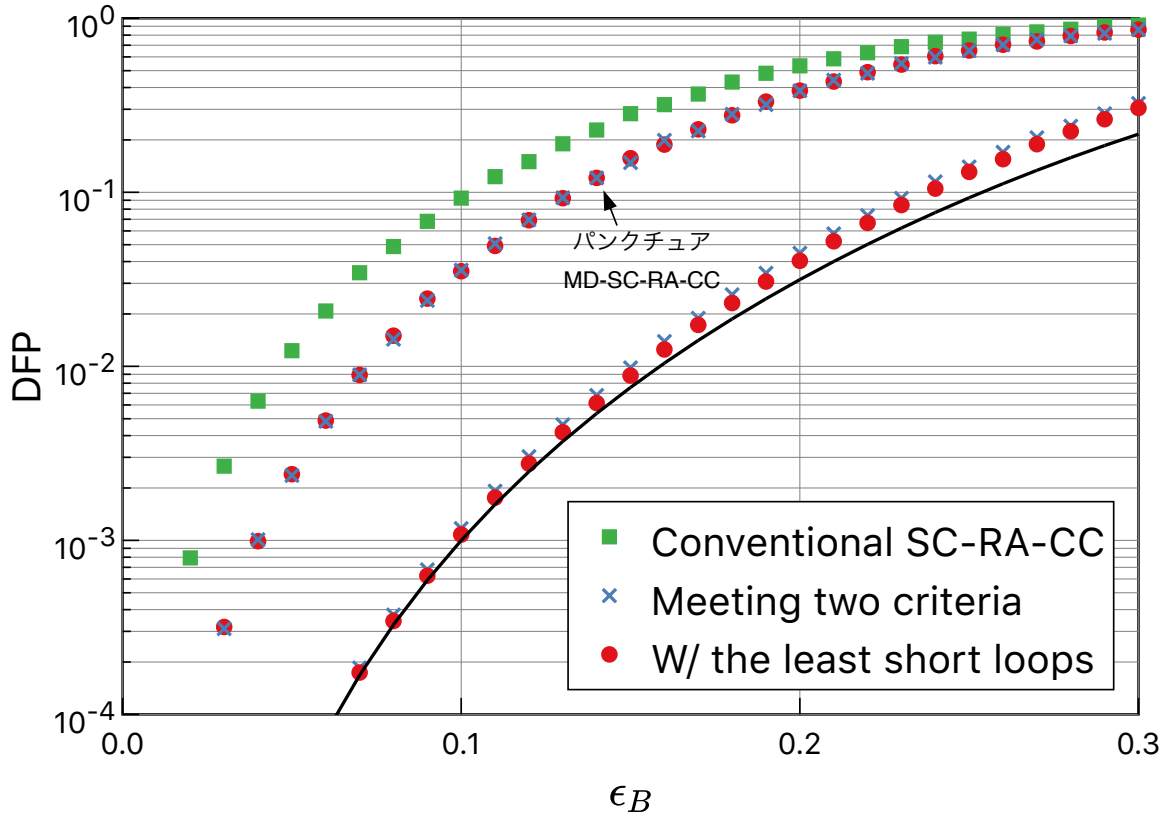


図 4.3: 1-BEC における結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, $(1, 2, 3, 4)$, $(1, 3, 4, 9)$ のパンクチャド MD-SC-RA-CC の DFP 特性

4.6.2 1-BEC の場合

図 4.3 にパンクチャド MD-SC-RA-CC の 1-BEC 上における DFP 特性を示す．結合ベクトルは $(1, 2, 3, 4)$, $(1, 3, 4, 9)$, $(2, 6, 8, 9)$ を用い，パラメータは端末数 $N = 5$ ，ブロック数 $T = 20$ ，ベクトルサイズ $Q = 4$ とし，全ての符号化率は $r = 0.493$ である．図 4.3 からパンクチャド MD-SC-RA-CC の DFP 特性は下界から，約 0.07 分だけ左にシフトすることがわかる．同様に，同じ値だけパンクチュアされていない MD-SC-RA-CC の DFP 特性からも劣化している．しかし，SC-RA-CC の DFP 特性と比較すると，同じ符号化率に関わらず，約 0.2 分だけ右に位置しており，パンクチャド MD-SC-RA-CC の方が DFP 特性において優れていることがわかる．これは，パンクチュアによる影響よりも小ループによる DFP 特性の劣化の方が大きいことを意味している．

4.6.3 T -BEC の場合

表 4.4 と表 4.5 に，結合ベクトル $(1, 2, 3, 4)$, $(1, 3, 4, 9)$, $(2, 6, 8, 9)$, $(1, 2, 3, 4, 5)$, $(1, 3, 6, 10, 11)$, $(1, 4, 8, 9, 11)$ によって与えられるパンクチャド MD-SC-RA-CC の DFP 特性を示す．端末数 $N = 5$

表 4.4: T -BEC における結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$, $(1, 2, 3, 4)$, $(1, 3, 4, 9)$ パンクチャド MD-SC-RA-CC の DFP 特性 ($N = 5$, $T = 20$, $Q = 4$)

\mathbf{v}	SC-RA-CC (1, 2, 3, 4)	パンクチャド MD-SC-RA-CC (1, 3, 4, 9)	下界 (2, 6, 8, 9)
Coding rate	0.493	0.493	0.493
DFP	4.50×10^{-2}	8.56×10^{-3}	8.56×10^{-3}
E_0	0	0	0
E_1	0	0	0
E_2	5	0	0
E_3	10	10	10
E_4	5	5	5
E_5	1	1	1

のときは $r = 0.493$, $N = 6$ のときは $r = 0.489$ とし, ブロック消失確率は $\epsilon_B = 0.10$, DFP 特性の下界も同時に示す.

表 4.4 の結果は表 4.1 の結果と一切変わっていないと言える. これは, 下界における最大の瞬時通信路容量と, MD-SC-RA-CC の符号化率との差が大きいため, パンクチュアリングによる MD-SC-RA-CC の符号化増加が DFP 特性に影響を与えていないと言える. ここから, 上記のギャップが大きい場合においては, パンクチャド MD-SC-RA-CC は SC-RA-CC よりも優れた方式であると言える.

表 4.5 に下界における最大の瞬時通信路容量と, MD-SC-RA-CC の符号化率との差が小さい場合の数値結果を示す. 表 4.2 と比較すると, $(1, 3, 6, 10, 11)$, $(1, 4, 8, 9, 11)$ の DFP 特性が下界から離れていることがわかる. 具体的には, $E_3 = 20$ が追加されており, 6 個のうち, 3 個の通信路が消失する全ての状態において復号が停止すると言える. さらに, $(1, 3, 6, 10, 11)$, $(1, 4, 8, 9, 11)$ の DFP 特性は等しくなっており, パンクチュアリングすることにより, フルエッジ結合の効果が現れなくなっていることがわかる. これは, 下界における最大の瞬時通信路容量と, MD-SC-RA-CC の符号化率との差が小さく, パンクチュアリングによる符号化率増加による影響が顕著に現れたためである. しかし, 等しい符号化率である SC-RA-CC と比較したとき, パンクチャド MD-SC-RA-CC は $E_2 = 0$ であるため, DFP 特性としては優れていると言える.

表 4.5: T -BEC における結合ベクトル $\mathbf{v} = (1, 2, 3, 4, 5), (1, 3, 6, 10, 11), (1, 4, 8, 9, 11)$ パンクチャード MD-SC-RA-CC の DFP 特性 ($N = 6, T = 15, Q = 5$)

\mathbf{v}	SC-RA-CC (1, 2, 3, 4, 5)	パンクチャード MD-SC-RA-CC		下界
		(1, 3, 6, 10, 11)	(1, 4, 8, 9, 11)	
Coding rate	0.489	0.489	0.489	
DFP	5.38×10^{-2}	1.59×10^{-2}	1.59×10^{-2}	1.27×10^{-3}
E_0	0	0	0	0
E_1	0	0	0	0
E_2	4	0	0	0
E_3	20	20	20	0
E_4	15	15	15	15
E_5	6	6	6	6
E_6	1	1	1	1

第5章 数値結果

本章では、2つの設計指針を満たす結合ベクトルについて1-ブロックとT-ブロックレイリーフェーディング通信路上での誤り率特性を計算機シミュレーションによって求める。評価指標としては、情報パケットのうち1ビットでも誤っているときに誤りとする、パケット誤り率 (PER: Packet Error Rate) を用いる。シミュレーションパラメータは表5.1に記載する。S-ランダムインターリーバのSパラメータは7とする。確率伝搬法としては、Sum-Product 復号法を用い、最大繰り返し回数を120回とした。本シミュレーションで用いる結合ベクトルは、できる限り小ループが少ないプロトグラフを構成する結合ベクトル $\mathbf{v} = (1, 3, 4, 9)$ と、フルエッジ条件を満たし、かつその中で最も小ループが少ないプロトグラフを構成する結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$ 、そして、従来方式のSC-RA-CCを構成する結合ベクトル $\mathbf{v} = (1, 2, 3, 4)$ である。上記2つのMD-SC-RA-CCはパンクチュアされ、3つの方式の符号化率は $r = 0.493$ に統一されている。

図5.1に、1-ブロックレイリーフェーディング通信路におけるパンクチャドMD-SC-RA-CCとSC-RA-CCのPER特性を示す。横軸は E_b/N_0 であり、 E_b は情報1ビットあたりの受信信号エネルギーを表す。図から、パンクチャドMD-SC-RA-CCの方がSC-RA-CCよりも $PER = 10^{-4}$ において、約1.5dB優れていることがわかる。これは図4.3の結果と対応していることがわかり、MD-SC-RA-CCを設計する場合、プロトグラフ内の小ループを少なくすることの重要性が1-ブロックレイリーフェーディング上においても示された。

図5.2に、T-ブロックレイリーフェーディング通信路におけるパンクチャドMD-SC-RA-CCとSC-RA-CCのPER特性を示す。図から、 $E_b/N_0 = [10, 16]$ において、同じ符号化率の3つの方式がほぼ同じPER特性を示すことがわかる。しかし、高SNR領域である $E_b/N_0 = [17, 20]$ においては、フルエッジ条件を満たし、かつその中で最も小ループが少ないプロトグラフを構成する結合ベクトル $\mathbf{v} = (2, 6, 8, 9)$ がSC-RA-CCの $\mathbf{v} = (1, 2, 3, 4)$ よりも、約0.5dB優れていることがわかる。これは図??の結果と対応しており、MD-SC-RA-CCを設計する場合、フルエッジ条件を満たしかつプロトグラフ内の小ループをできる限り少なくすることの重要性がT-ブロックレイリーフェーディング上においても示された。

表 5.1: シミュレーションパラメータ

		パンクチャド MD-SC-RA-CC	SC-RA-CC
結合ベクトル	\mathbf{v}	(1, 3, 4, 9)	(1, 2, 3, 4)
符号化率	r	0.493	
端末数	N	5	
ブロック数	T	20	
ベクトルサイズ	Q	4	
パケットサイズ [bits]	M	100	

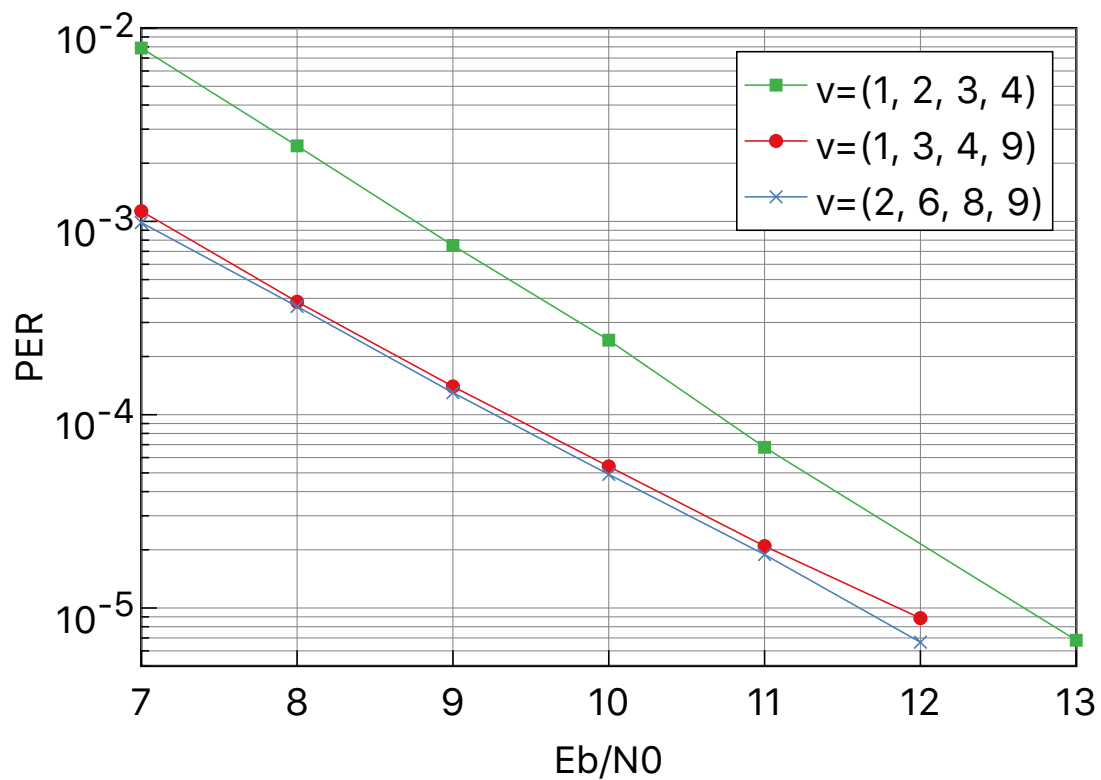


図 5.1: 1-ブロックレイリーフェーディング通信路における結合ベクトル $\mathbf{v} = (1, 3, 4, 9)$, $(2, 6, 8, 9)$, $(1, 2, 3, 4)$ であるパンクチャド MD-SC-RA-CC と SC-RA-CC の PER 特性

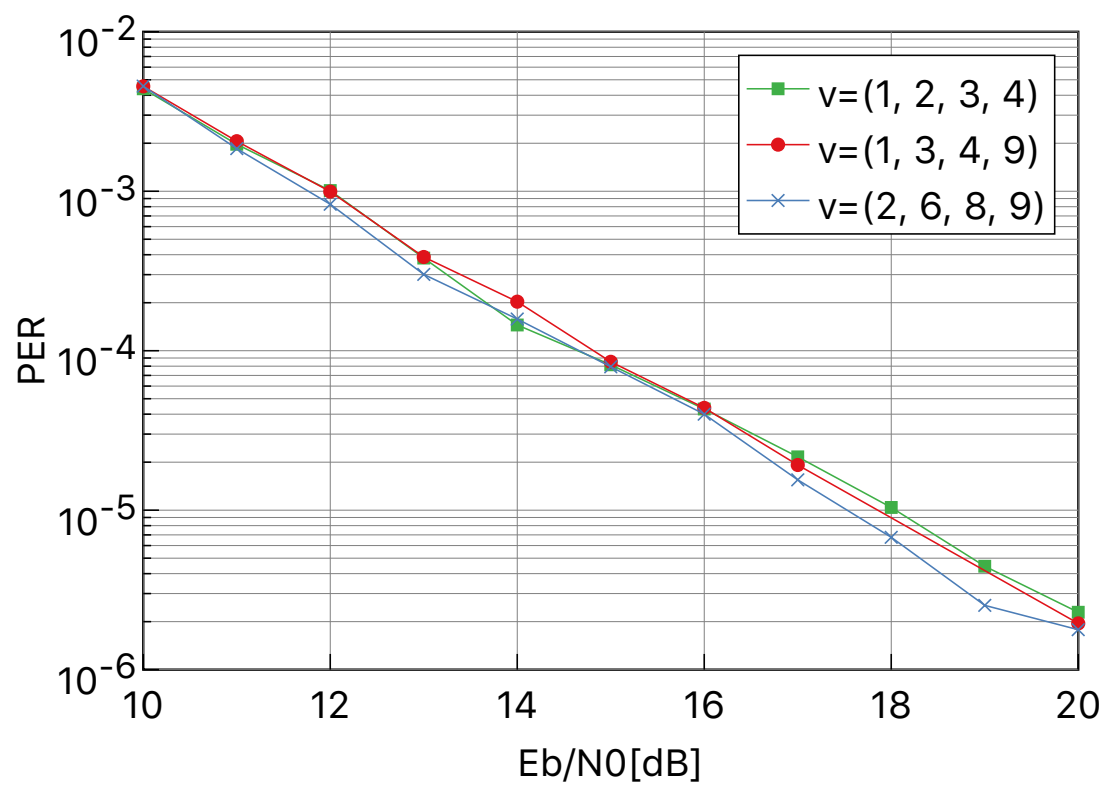


図 5.2: T -ブロックレイリーフェーディング通信路における結合ベクトル $\mathbf{v} = (1, 3, 4, 9)$, $(2, 6, 8, 9)$, $(1, 2, 3, 4)$ であるパンクチャド MD-SC-RA-CC と SC-RA-CC の PER 特性

第6章 結論

本研究において、優れた復号特性を示す MD-SC-RA-CC の符号化構造を決定する結合ベクトルの導出方法を提案した。簡易化した通信路モデルである 1-BEC と T -BEC に基づき、2つの設計指針を導出した。1つ目はプロトグラフ上の小ループをできるだけ少なくすることであり、2つ目は情報パケットとそれを符号化したパリティパケットがお互いに統計的に独立な通信路を伝搬している状態であるフルエッジ結合条件である。BEC 上における数値結果から、上記2つの設計指針を満たすパンクチャド MD-SC-RA-CC の DFP 特性は SC-RA-CC よりも優れていることが明らかになった。最後に、1-ブロックと T -ブロックレイリーフェーディング通信路上において、符号化率を揃えた状態で、提案した方式のパンクチャド MD-SC-RA-CC と従来方式の SC-RA-CC の誤り率特性を計算機シミュレーションによって評価した。1-ブロックレイリーフェーディング通信路では、パンクチャド MD-SC-RA-CC が SC-RA-CC よりも優れた復号特性を示していることが明らかになった。 T -ブロックレイリーフェーディング通信路においても同様に、パンクチャド MD-SC-RA-CC は SC-RA-CC よりも優れた復号特性を示すことがわかった。以上から、ブロックレイリーフェーディングのような通信路変動が発生する環境に対しては、SC-RA-CC よりもパンクチャド MD-SC-RA-CC の方が、ロバストであり、より優れた通信方式である。

謝辞

初めに，本論文の執筆，及びこれまでの研究活動にあたり，非常に多くの助言を賜りました石橋功至准教授に深く感謝いたします。加えて，研究内容に関して有意義な議論を交わして下さいた電気通信大学 AWCC の山尾泰教授，藤井威生教授，安達宏一准教授にも感謝申し上げます。シンガポール化学技術研究庁 (A*STAR) のインフォコム研究所 (I²R)，Dr. Sumei Sun には留学における支援，そして研究に関する洞察に満ちた数々の助言をありがとうございました。石橋研究室を始め，山尾研究室，藤井研究室，安達研究室のメンバーの皆様とも，日頃から相談や議論を行うことができたことは私の大きな糧です。最後に，ここまで経済的，精神的に支えてくれた家族に心より感謝の意を表します。

関連発表

- R. Tanaka and K. Ishibashi, “Robust coded cooperation based on multi-dimensional spatially-coupled repeat-accumulate codes,” in *Proc. IEEE WCNC*, San Francisco, CA, Mar. 2017.
- 田中亮輔, 石橋功至, “レイリーフェーディング環境下における多次元空間結合 Repeat-Accumulate 符号化協調のパケット誤り率特性評価,” 電子情報通信学会 2016 ソサイエティ大会, 北海道大学, vol. B-5-95, Sep. 2016.

参考文献

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.
- [2] S. M. Alamouti, “A simple transmit diversity technique for wireless communications,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.
- [3] A. Nosratinia, T. E. Hunter, and A. Hedayat, “Cooperative communication in wireless networks,” *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 74–80, Oct. 2004.
- [4] J. N. Laneman, G. W. Wornell, and D. N. C. Tse, “An efficient protocol for realizing cooperative diversity in wireless networks,” in *Proc. IEEE Int. Symp. on Inform. Theory*, 2001, pp. 294–.
- [5] A. Sendonaris, E. Erkip, and B. Aazhang, “User cooperation diversity. Part I. System description,” *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.
- [6] —, “User cooperation diversity. Part II. Implementation aspects and performance analysis,” *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939–1948, Nov. 2003.
- [7] T. E. Hunter and A. Nosratinia, “Cooperation diversity through coding,” in *Proc. IEEE Int. Symp. on Inform. Theory*, 2002, pp. 220–.
- [8] —, “Diversity through coded cooperation,” *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 283–289, Feb. 2006.
- [9] M. Janani, A. Hedayat, T. E. Hunter, and A. Nosratinia, “Coded cooperation in wireless communications: space-time transmission and iterative decoding,” *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 362–371, Feb. 2004.
- [10] X. Bao and J. Li, “Adaptive network coded cooperation (ANCC) for wireless relay networks: matching code-on-graph with network-on-graph,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 574–583, Feb. 2008.
- [11] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, 1963.
- [12] T. Richardson, “Error floors of LDPC codes,” in *41st Annu. Allerton Conf. on Commun. Control and Computing*, Oct. 2003.

- [13] K. Zhang, L. Yin, Y. Wang, and J. Lu, “Optimized relay selection strategy for adaptive network coded cooperation,” in *21st Annu. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun.*, Sep. 2010, pp. 2309–2313.
- [14] S. Johnson and G. Lechner, “Spatially coupled repeat-accumulate codes,” *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 373–376, Feb. 2013.
- [15] A. J. Felstrom and K. S. Zigangirov, “Time-varying periodic convolutional codes with low-density parity-check matrix,” *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
- [16] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [17] S. Kudekar, T. J. Richardson, and R. L. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [18] S. Kudekar, T. Richardson, and R. L. Urbanke, “Spatially coupled ensembles universally achieve capacity under belief propagation,” *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.
- [19] R. M. D. Divsalar, H. Jin, “Coding theorems for turbo-like codes,” in *36th Allerton Conf. on Commun. Control and Comput.*, 1998, pp. 201–210.
- [20] N. Takeishi and K. Ishibashi, “Spatially coupled repeat-accumulate coded cooperation,” in *IEEE Wireless Commun. and Networking Conf.*, Mar. 2015, pp. 516–521.
- [21] R. Ohashi, K. Kasai, and K. Takeuchi, “Multi-dimensional spatially-coupled codes,” in *IEEE Int. Symp. on Inform. Theory*, July 2013, pp. 2448–2452.
- [22] 武石直樹, 石橋功至, アブレウジュゼッペ, “分散多次元空間結合 Repeat-Accumulate 符号,” *信学技報*, vol. 114, no. 490, pp. 574–583, Mar. 2015.
- [23] 和田山正, 低密度パリティ検査符号とその復号法. トリケップス, Jun. 2002.
- [24] H. Jin, “Analysis and design of turbo-like codes,” Ph.D. dissertation, California Institute of Technology, Pasadena, California, May 2001.
- [25] 和田山正, 誤り訂正技術の基礎. 森北出版株式会社, 2010.
- [26] B. Vucetic and J. Yuan, *TURBO CODES Principles and Applications*. Kluwer Academic Publishers, 2000.
- [27] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.

- [28] A. G. i Fabregas, “Coding in the block-erasure channel,” *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5116–5121, Nov. 2006.
- [29] J. A. McGowan and R. C. Williamson, “Loop removal from LDPC codes,” in *Proc. IEEE Inform. Theory Workshop*, Mar. 2003, pp. 230–233.